

Heuristics for Mitigating Mode Confusion in Digital Cameras

by

Victor Vui-Kiat Chong

B.A., University of Victoria, 2000

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in the Department of Computer Science

Copyright © 2006 by Victor Vui-Kiat Chong
University of Victoria

*All rights reserved. This work may not be reproduced in whole or in part, by
photocopy or any other means, without the permission of the author.*

Heuristics for Mitigating Mode Confusion in Digital Cameras

by

Victor Vui-Kiat Chong

B.A., University of Victoria, 2000

Supervisory committee

Dr. Margaret-Anne D. Storey (Department of Computer Science)

Supervisor

Dr. Hausi A. Müller (Department of Computer Science)

Departmental Member

Dr. Daniel M. German (Department of Computer Science)

Departmental Member

Dr. Nigel J. Livingston (Department of Biology)

External Examiner

Supervisory committee

Dr. Margaret-Anne D. Storey (Department of Computer Science)

Supervisor

Dr. Hausi A. Müller (Department of Computer Science)

Departmental Member

Dr. Daniel M. German (Department of Computer Science)

Departmental Member

Dr. Nigel J. Livingston (Department of Biology)

External Examiner

Abstract

Mode confusion describes the psychological condition of a human operator when he or she finds it difficult or impossible to determine the current state of an automated system. The ability for a user to keep track of these modes internally is especially important in systems where the number of available functions is greater than the number of controls or indicators present in the user interface.

In this thesis, we present a set of heuristics, intended for use by engineers during design time, to mitigate mode confusion potential in digital cameras. To test the applicability and effectiveness of these heuristics, we simulated a formative evaluation, as would be performed by a designer, and correlated these results with that of a usability study.

Table of contents

Supervisory committee	ii
Abstract	iii
Table of contents	iv
Table of figures	viii
Acknowledgements	x
Dedication	xii
1. Introduction	1
1.1 Research motivation, goals, and question.....	2
1.2 Thesis outline.....	5
2. Human-computer interaction	6
2.1 What is human-computer interaction?.....	6
2.2 Norman's principles of design.....	7
2.2.1 Affordances.....	8
2.2.2 Visibility.....	8
2.2.3 Feedback.....	9
2.2.4 Mapping.....	10
2.2.5 Constraints.....	10
2.3 Errors of commission and omission.....	11
2.4 Gulfs of execution and evaluation.....	12
2.5 Usability testing.....	13
2.5.1 Formative evaluations.....	14
2.5.2 Summative evaluations.....	14
2.6 Predictive evaluations.....	14

2.6.1	Cognitive walkthroughs.....	14
2.6.2	Heuristic evaluations	15
2.7	User studies.....	16
	Chapter summary.....	17
3.	Mode confusion	18
3.1	What is a mode?.....	18
3.2	Why do we use modes?.....	19
3.3	What are the different types of modes?	20
3.3.1	Interface modes.....	20
3.3.2	Functional modes.....	24
3.3.3	Supervisory modes	25
3.4	What is mode confusion?.....	27
3.5	Why is the study of mode confusion important?	28
3.6	Modes and mental models	29
3.7	Why do good mental models go bad?.....	32
3.7.1	Errors in interface interpretation.....	32
3.7.2	Inconsistent system behaviour.....	35
3.7.3	Indirect mode changes	36
3.7.4	Unclear operator authority limits.....	37
3.7.5	Unintended side-effects	38
3.7.6	Lack of appropriate feedback	39
3.8	Requirements engineering and mode confusion	41
	Chapter summary.....	41
4.	A set of heuristics for mitigating mode confusion	43
4.1	Why is it important to develop these heuristics?	43
4.2	For the designer: a note before using these heuristics	46
	What is a heuristic evaluation?	46
	How is this technique different from Nielsen's usability heuristics?	47
	Should I use these heuristics instead of performing a user study?.....	47
	How should I use these heuristics?	47
	What should I avoid doing?	48
4.3	For the designer: a set of heuristics for mitigating mode confusion.....	48
	A: Visibility	48

A1: Give important features more visibility	49
A2: Associate frequently used functions with frequently used buttons.....	49
A3: Associate infrequently used functions with infrequently used buttons	50
A4: Use unique adornments to differentiate between interface modes	51
B: Feedback	52
B1: Let the user know what is happening.....	52
B2: Let the user know what is going to happen next.....	53
B3: Use indicators to show non-default behaviour.....	54
B4: Use alerts and dialogues judiciously.....	54
C: Constraints.....	55
C1: Prohibit a single control from changing more than one variable	55
C2: Make potentially hazardous functions harder to activate.....	56
D: Consistency	57
D1: Distinguish between features that can and cannot be performed.....	57
D2: Maintain a consistent spatial layout within modes	58
D3: Use only one menu design paradigm per interface.....	59
D4: Ensure that related operations behave in similar ways	60
Chapter summary	61
5. Application of the heuristics to digital cameras: heuristic evaluation	62
5.1 Research goal	62
5.2 Research question	65
5.3 Evaluation process	65
5.3.1 Collating a set of general use cases for digital cameras	66
UC1: Turning the camera on and off.....	67
UC2: Taking a picture	67
UC3: Using scene modes.....	68
UC4: Reviewing photographs	68
UC5: Deleting a single photograph and deleting all photographs.....	69
5.3.2 Selecting digital cameras used in the evaluation	69
5.3.3 Description of digital cameras.....	69
Canon PowerShot A40	70
Canon PowerShot S30.....	71
Olympus Camedia C-3040 Zoom.....	72
5.4 Heuristic evaluation	72
5.4.1 Recruitment of an expert user as “evaluator”	73
5.4.2 Formative evaluation of the heuristics.....	73

5.4.3	Heuristic evaluation results	74
	Canon PowerShot A40	74
	Canon PowerShot S30	77
	Olympus Camedia C-3040 Zoom	78
5.4.4	Analysis of heuristic evaluation results	79
	Chapter summary	81
6.	Application of the heuristics to digital cameras: usability study	82
6.1	A set of extended use cases for exposing specific confusion scenarios	83
	EUC1: Determining whether the PowerShot A40 is powered on or off	83
	EUC2: Changing focus modes on the Camedia C-3040 Zoom	84
	EUC3: Changing between shooting modes on the PowerShot S30	84
	EUC4: Determining if the Camedia C-3040 Zoom is in standby mode	84
	EUC5: Using the image protection feature on the PowerShot S30	85
6.2	Recruitment of participants	85
6.3	Participants dossiers	86
6.4	Procedure	89
6.5	Usability study results	91
	Chapter summary	94
7.	Future work and contributions	96
7.1	Discussion and future work	96
7.2	Research contributions	99
7.3	Epilogue	100
	References	101
	Appendix A: Certificate of approval	107
	Appendix B: Questionnaire	108
	Appendix C: Mental models of Canon PowerShot S30 sliding lens cover design	111

Table of figures

1.1	Fujifilm FinePix 6800 Zoom.....	3
2.2.1	Different affordances of two door designs.....	8
2.2.2	Power button of Macintosh PowerBook	9
2.2.3	Warning lights used as feedback annunciators	9
2.2.4	Throttle quadrant of Boeing 747 airliner	10
2.2.5	Film door lock release button of Nikon F100 film camera.....	11
2.4	Button panel and “earthquake” button from elevator	13
3.3.1a	Adaptive user interface in Microsoft Word	21
3.3.1b	An example of input type checking	22
3.3.1c	An expanding hierarchical tree	23
3.3.3	Confirmation dialogue from Gateway DC-T50 digital camera	27
3.5	Size comparison of an early mobile telephone with a contemporary design	29
3.6	Norman’s mental model of his refrigerator.....	31
3.7.1a	Annunciator duality in inkjet printer design	33
3.7.1b	Digital clock with inconspicuous indicator to show mode	34
3.7.3	Label from paper shredder showing intended duty cycle	36
3.7.4	Schematic diagram of a typical automobile anti-lock breaking system.....	38
3.7.5	System model of Norman’s refrigerator	39
4.1	Common entry door handle from a North American home	44
A1	Pentax Optio S5n: a case of form over function?	49
A2	Shutter release button from Nikon D200 digital camera	50
A3	Nikon D70’s top-deck LCD panel illumination button.....	51
A4	Colour-coded title bars and icons from HP PhotoSmart R707	52

B1	Viewfinder display from Nikon D2Hs.....	53
B2	E-commerce checkout progress bar	53
B3	Control panel from dishwasher with two, four, or six hour delay function	54
B4	Kodak DC265 Zoom: memory card formatting confirmation dialogue	55
C1	“Reset” button on photocopier.....	56
C2	Confirmation dialogue screen from BlackBerry 7130c wireless handheld.....	57
D1	Use of illumination to indicate key guard on Nokia 3100 mobile telephone.....	58
D2	An example of a disable function in Canon PowerShot A700.....	59
D3	Example of a consistent user interface used by many Sony products.....	60
D4	Image of Canon 24-70mm f/2.8L USM lens	61
5.3.3a	Canon PowerShot A40.....	70
5.3.3b	Canon PowerShot S30	71
5.3.3c	Olympus Camedia C-3040 Zoom	72
A	User study: certificate of approval	107
C.1	System model of Canon PowerShot S30	112
C.2a	Mental model of Participant A.....	113
C.2b	Mental model of Participant B.....	113
C.2c	Mental model of Participant C.....	114
C.2d	Mental model of Participant D.....	114
C.2e	Mental model of Participant E.....	114
C.2f	Mental model of Participant F.....	115
C.2g	Mental model of Participant G.....	115

Acknowledgements

While a thesis may be a solitary pursuit, it is by no means a solitary effort. During the course of my research, I have had the pleasure of working with and seeking the help of many remarkable individuals who have shared with me their knowledge and insight. In particular, I would like to thank the following people, without whom this thesis could not have been realized:

Margaret-Anne Storey, for her encouragement, guidance and friendship. As a mentor and supervisor, Dr. Storey has been instrumental in my academic development and the production of the volume in your hands. I have benefited greatly from her insightful criticisms, been buoyed by her steadfast faith in my abilities (even when my own conviction wavered), and have been nurtured by her well-tempered approach to education that strikes a perfect balance between firmness and compassion.

Hausi Müller, for getting the ball rolling. It is very fitting that Dr. Müller should be part of my supervisory committee as he was the lecturer for my first class at the University of Victoria ten years ago. It was his enthusiasm and encouragement early on in my academic career – both inside and outside of the classroom – that convinced me to pursue the study of Computer Science.

Daniel German for his scientific and artist insights. Though his favourite “team” wears a red jersey and mine a yellow, I have a great respect for his work and his wiliness to share his expertise in the fields of computing and photography.

The members of the **Computer Human Interaction & Software Engineering Laboratory** (CHISEL). My tenure in the Department of Computer Science would have been possibly less educational – but certainly less enjoyable – had it not been for the high-spirited antics of my fellow group members, both past and present. I will always remember CHISELers for their off-the-wall deportment during leisure time but cut-throat acuteness during oral presentations.

And most importantly, my father and mother, **Michael** and **Louisa**, for their unbridled encouragement and support. I am one of those individuals fortunate enough to have parents who are willing to give their progeny every advantage and opportunity they never had themselves.

And now, for something completely different.

Dedication

To my mother and father, Louisa and Michael.

1

Introduction

Mode confusion, succinctly, is a term used to describe the psychological state of a human operator when he or she finds it difficult or impossible to determine the current mode of an automated system. The ability for a user to keep track of these modes internally, or with the assistance of cognitive aids (such as visual cues or audible alerts), is especially important in systems where the number of available functions are greater than the number of controls or indicators present in the user interface. In such situations, the personality of these controls and indicators change based on the context of the system and it is important that users are able to determine the operating mode of the automation in order to anticipate the system's behaviour.

One obvious solution to this example of mode confusion would be to provide an input for every possible function in a system and an indicator for every output, but this strategy is only practical in simple devices. The current design trend is to make systems as feature-laden and miniature as possible [18, 19], necessitating the use of mode-based arrangements, which have the undesirable side effect of imposing greater cognitive demands upon users. Even if we accept that modes are a necessary component of complex systems, it is desirable nevertheless to construct designs that attenuate the potential for mode confusion as much as possible.

This thesis makes one critical assumption: that the application of theories from human-computer interaction will result in a more usable product. Note that we have been careful not

to use the word useful, as the utility of a product is determined by its functionality, whereas usability is governed by its design. Of course, these two factors relate strongly to each other because a system that is unusable is useless, and a usable system that does nothing is likewise ineffectual [17].

1.1 Research motivation, goals, and question

Mitigation of mode confusion is not a feature that can be grafted onto a system but must be integrated into a design's philosophy from first inception. To date, a limited amount of research has been done to establish design guidelines to help control mode confusion [11, 44]. Most research conducted in this field explores the subject at the cognitive and physiological level, providing many theories as to why mode confusion occurs [8, 13], but offers little advice to the designer on how to avoid designs with mode confusion potential. Other researchers are promoting the use of automated methods [3, 25, 35, 39] to identify mode confusion through model checking algorithms, but again these procedures only yield identification, not solution.

As it has only been in recent years that particular emphasis has been placed on teaching user-centred development in classrooms [47], established software engineers may be unfamiliar or unconvinced of the value of user-centred design [21]. Converting third-year software engineering students into practitioners of human-computer interaction may not be easy as the nature of programming emphasizes functionality over form, and utility over usability. Having identified this dearth of experts in human-computer interaction, we were interested to know if there was a method through which we could make the theories of human-computer interaction more accessible to non-experts without inundating designers with paragraphs on cognitive theory, information visualization, and interaction design. Indeed, any attempt involving a formal discussion of these theories may be more akin to a condensed course in first principles, so we instead sought to develop a set of heuristics for expert and non-expert

designers alike to help mitigate mode confusion. With this research goal in mind, we began to formulate a set of heuristics that a designer could conceivably use as a guideline to develop a level of basic usability that excludes the most common pitfalls in ill-conceived interface designs. Once a set of heuristics was developed, our research question was then to determine whether users without an expertise in human-computer interaction would be able to apply the heuristics and positively identify scenarios of mode confusion potential.

Developing a set of generalized heuristics may have been inadequate as such a solution would have been applicable to many scenarios at a rudimentary level, but insufficient to solve specific usability issues, with respect to mode confusion potential, in a given domain without further extrapolation. We therefore chose to formulate heuristics specifically to mitigate mode confusion potential in digital cameras. Like mobile telephones, the “consumerization” of digital cameras has now entered the domain of fashion – where aesthetic factors take precedence over both utility and usability – and many recent models have sleek lines and great aesthetic appeal but are plagued by notoriously difficult-to-use interfaces.



Figure 1.1: The Fujifilm FinePix 6800 Zoom's exterior was designed by the famed F.A. Porsche design group, the same firm that has been tapped to design first-tier coffee makers, toasters, vegetable knives, hard drive enclosures... and automobiles.

Digital photography was selected as our research domain for two reasons. Firstly, digital imaging is a technology that is currently enjoying remarkable market growth and development. While many other product categories have seen a slowdown or moderation of innovation during recent times, manufacturers of digital cameras continue to push the boundaries of the socially familiar and technically possible with each successive product generation. As digital cameras are an emerging technology – even more than ten years after their introduction – conventions regarding the operational behaviour and technical ability of the “quintessential” digital camera user interface, against which other designs are to be judged, have yet to be established. Because no one design paradigm has presented itself as the undisputed preference of users, manufacturers continue to introduce a wide variety of user interface styles in search of the proverbial “silver bullet.” This impetus to bring new designs to fruition makes the digital camera market a very dynamic and engaging field of study.

Secondly, as film-based photography is a well established and mature technology, the move to digital technologies is particularly interesting because camera manufacturers must acknowledge – and in some ways emulate – the entrenched and widely-accepted design principles and interaction styles of film-based cameras even if these techniques are less-than-ideal for digital technologies. Striking a good balance between familiarity and innovation is therefore important to designers who are trying to convince users of film to transition to the digital domain. Even for existing digital camera users, this need to maintain “legacy interaction” is important and designers must acknowledge that users with a proven familiarity and demonstrated proficiency in an existing design may be unwilling to invest any significant effort to learn the operation of a new system.

1.2 Thesis outline

Our set of heuristics to mitigate mode confusion potential in digital cameras are grounded in the study of human-computer interaction and §2 discusses some of the most salient principles that directly contributed to the formation of these heuristics. §3 focuses specifically on the issue of mode confusion, a type of user error that occurs frequently in complex designs [22]. The chapter begins with an expansion of the definition of mode confusion and follows with a dossier of cognitive factors that can lead to mode confusion potential.

§4, §5, and §6 form the core contribution of this thesis. §4 recounts the development of the heuristics from the distillation of fundamental human-computer interaction principles and presents the set of heuristics as a deliverable that is intended for use by digital cameras designers who are experts or who have little or no experience in the field of human-computer interaction. §5 and §6 documents an evaluation process, employing a heuristic evaluation and user study, to determine whether the heuristics could be successfully applied by a non-expert user to uncover verifiable instances of mode confusion potential. A discussion of the results from both the heuristic evaluation and user study are also included in §6.

§7 provides a summation of the material presented in this thesis, discusses the contributions of our research, and identifies several opportunities for potential future work.

2

Human-computer interaction

This chapter has two main goals: to discuss aspects of human-computer interaction that are relevant to the study of mode confusion; and to define several techniques commonly used to evaluate the usability of user interfaces. The first half of this chapter includes a general introduction to the field of human-computer interaction (§2.1), an overview of Normans' principles of design (§2.2), a discussion of errors of commission and omission (§2.3), and a review of the gulfs of execution and evaluation (§2.4). The second half of the chapter focuses on methods of usability testing and predictive evaluations: §2.5 discusses the purpose of usability testing, how it is helpful in determining usability problems, and describes formative and summative evaluations; §2.6 introduces several predictive evaluation techniques; and §2.7 compares and contrasts two different, but complementary, usability testing techniques: heuristic evaluations and user studies.

2.1 What is human-computer interaction?

The study of *human-computer interaction*, also commonly referred to as *human-machine interaction* or *human factors*, is devoted to understanding how humans interact with machines [9]. Application of this research enables designers to develop user-centered interfaces that attempt to maximize the user's ability to control or an automated system. Different designs call on different strategies and depending on the complexity of the task, some designs, such as those tailored for use during emergency situations, may concentrate on using natural mappings to take advantage of human intuition, while other designs such as

those found in the cockpits of advanced commercial aircraft, strive to increase the density of functionality within a framework made familiar through explicit training and instruction [6]. The fluidity of such interaction relies heavily on this bi-directional flow that translates human intentions into machine instructions and machine feedback into meaningful indications to the operator.

Though a user interface is an integral part of any device designed for manipulation by a human, they are typically relegated to acting as interlocutor between human and machine [33]. At the highest level, most devices designed for human manipulation can be parceled into two portions consisting of functionality and interface. The core functionality incorporates the set of functions that allow the device to fulfil the purposes for which the device was designed while the functions, controls, and indicators of the user interface provide the operator with a means to monitor or manipulate the processes that are autonomically performed by the system. For example, a digital alarm clock can be divided into two sub-elements: the time keeping apparatus and the user interface. The time keeping portion consists of the diodes, transistors, and power supply – the components integral to a digital clock's timekeeping – while the user interface, with its illuminated digital numbers and the dials or buttons, provide a means of input and output. At its most essential, a chronometer need only be a device that accurately keeps time and thus a clock with no buttons and no display is still a chronometer; what makes an alarm clock accessible and useful to humans are the methods of access afforded by the user interface.

2.2 Norman's principles of design

With respect to the mechanics of human-computer interaction, Norman [30] has introduced a number of terms that successfully capture some of the most essential aspects of interaction design.

2.2.1 Affordances

An *affordance* is a property of an object that indicates how to interface with that object. Though the interface may allow for manipulation of the object in a number of ways, the experience of the user will typically dictate the specific way in which the interface is to be used. The purpose of the affordances perceived by the user may not be the same purposes as envisioned by the designer, however, especially if the design of the object is physically or logically similar to another object. In such a case, there is a potential for confusion of purpose or for an object to act as a surrogate for the other.

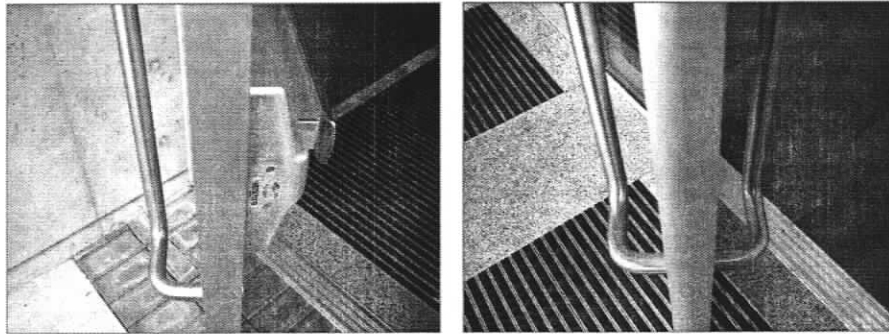


Figure 2.2.1: The vertical handle on one side of this door (L) affords pulling while the other has a large horizontal bar that affords pushing. In designs (R) where affordances have not been exploited or are used improperly, users may need to search for additional clues to determine the actual affordance. This door only opens one way but two vertical handles seems to imply that it should always be pulled.

2.2.2 Visibility

Visibility refers to the amount of exposure a feature is given. If a feature is highly visible, then it has been given a large degree of emphasis; if a feature is deemed to be of low visibility, then it has been marginalized or is completely hidden. Providing the right amount of visibility is important in human-computer interaction as having too many high-priority items can cause desensitization and information overload, while providing a feature with too little priority may mean that it may be missed, potentially leading to user error.



Figure 2.2.2: The power switch on this Apple PowerBook has been given greater visibility by placing it apart from the other buttons and keys. On some Macintosh models, the power light is also backlit for identification in darkened environments.

2.2.3 Feedback

Feedback is the process by which a system communicates with the user so that the user can maintain an awareness of the system's state. In ideal circumstances, the user would always know what the system was doing, be able to anticipate what the system was going to do next, consistently make the correct choices when given several options, and be able to take control if the automation behaved improperly.



Figure 2.2.3: These impressive-looking light fixtures let the user know that the laser is in operation. Not only do they warn users of radiation danger, but they also provide feedback to the user, showing that the laser has been engaged.

2.2.4 Mapping

Broadly, *mapping* refers to the relationship between two objects and in the field of human-computer interaction, refers specifically to the correlation between a physical control (or its virtual equivalent), its affordances, and the resulting action. Technically, a mapping merely suggests that there is a correlation between a control and a result effect and makes no claim as to the sensibility of the arrangement; the mapping may be confusing or counter-intuitive. A specific type of mapping, *natural mapping*, takes advantage of physical analogies and cultural standards to make mappings immediately understandable and approachable with little prior training or exposure.

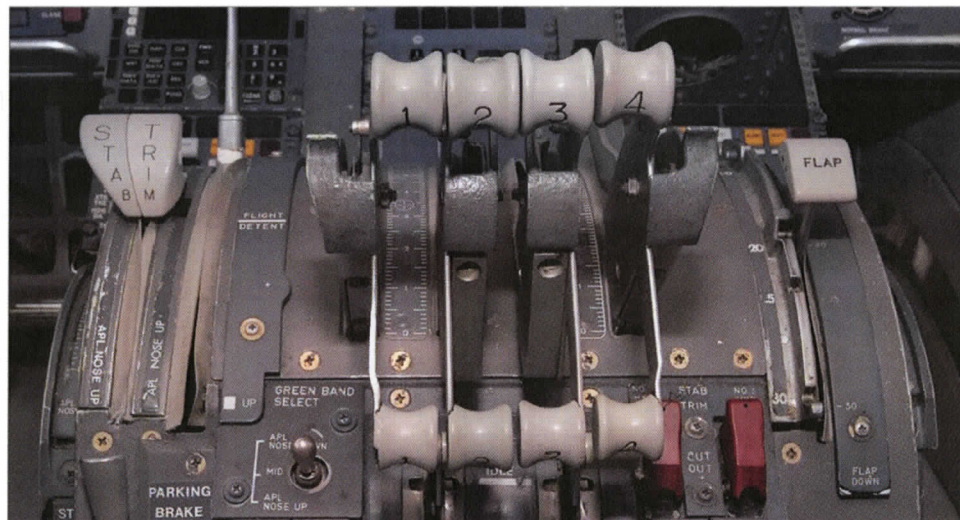


Figure 2.2.4: The four throttle levers of the Boeing 747 provide separate control over each of the airplane's four engines. From the pilot's vantage, Lever 1 controls the left outboard engine and Lever 2 controls the left inboard engine. Levers 3 and 4 use a similar mapping, but for the engines on the right-hand side of the airplane.

2.2.5 Constraints

By judiciously limiting the user's ability to affect change in a system, a designer can direct the user towards making a correct choice, taking an appropriate measure, or forcing the user to contemplate a potentially dangerous action before commitment. *Constraints* are key to good

user interface designs as they embed the knowledge of the designer within their structure; choices which the designer has deemed to be dangerous have been constrained so that a user cannot access them inadvertently.

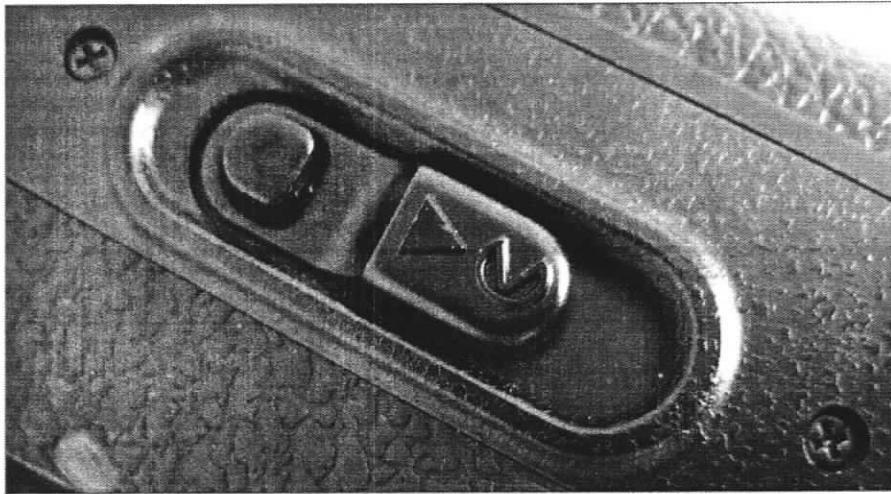


Figure 2.2.5: On the Nikon F100 SLR camera, the film door is protected from inadvertent opening by a lock release button. To open the film door, the user must depress the lock release button while simultaneously sliding the lock release lever.

2.3 Errors of commission and omission

First defined by Sarter and Woods [41], *errors of commission* and *errors of omission* are two types of user errors that relate to the inopportune action or inaction of the user. An error of commission occurs when a user performs an inappropriate action at the correct time, an inappropriate action at an incorrect time, or an appropriate action at an incorrect time. In contrast, an error of omission occurs when a user fails to perform an appropriate action at the correct time.

Errors of commission may be inadvertent or intentional. An errant command may be inadvertently passed to the automation without the intention of the operator if, for example, the operator slips while using a hand-held controller, depresses an actuator with an elbow, or

indiscreetly thinks aloud while using a voice-activated system. Errors of commission may also be done purposefully by the operator if the operator is led to believe – due to poor training, a misunderstanding of the task at hand, or unfamiliarity with the underlying automation – that the action they commissioned was both appropriate and timely.

Errors of omission are concerned with situations in which the inaction of the operator is in itself an error. In cases where the execution of a set of commands must be performed in a specific order or within some time constraint, the failure of action on the part of the user may lead to erroneous system behaviour. As many automated systems are designed to operate with a minimum of user interaction for reasons of efficiency, errors of omission are most common in non-standard situations where the automation has encountered a scenario from which it cannot automatically recover and is deferring to the operator to provide assistance or assume manual control. Errors of omission are a particularly insidious type of user error as they are caused by inaction; errors of omission may be difficult to identify because a user may believe that since they have not performed an action they have not contributed to the problem.

2.4 Gulfs of execution and evaluation

The *gulf of execution* describes the degree by which a user's perception of the capabilities and affordances offered by a system differ from those actually provided by the system. When the gulf of execution is small, a user's perception of how a system operates and the mechanisms by which to manipulate the system will closely resemble the actual system operation and manipulation methods. When the gulf of execution is large, there may be a total disconnect between the user's mental conception of the system and the actual model as developed by the designer. Reduction of the gulf of execution is desirable as a close match between the user's mental model and the actual system model allows the user to better understand the system in non-nominal situations and predict future system behaviour.

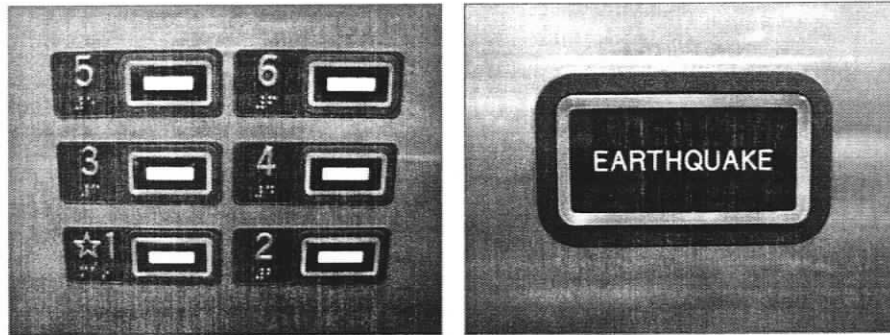


Figure 2.4: In this elevator, floors are selected by pressing the rectangular shaped buttons (L). When a floor is selected, the bezel of the button is illuminated in red. Located immediately above this panel is this peculiar item (R), which looks suspiciously similar to the other buttons and is labelled “Earthquake”. What action does this button afford? Luckily, this “button” cannot be pressed and the author can only assume that the bezel illuminates during an earthquake.

The *gulf of evaluation* describes the psychological gap that an operator must cross in order to interpret a user interface. In a system where the gulf of evaluation is small, the user interface takes advantages of usability principles to present information that is readily interpretable by the user and the system behaves in a manner consistent with how the user thinks the system should operate. In systems where the gulf of evaluation is large, the user is unable to properly interpret the user interface, preventing the user from properly determining the system’s state, the options available to the user, or the correct course of action.

2.5 Usability testing

Usability testing is a means for determining how well a user can use a system to fulfill its intended purpose. Usability testing is typically part of a larger iterative design process whereby problems in the system are identified, resolved and resubmitted for testing. Formative and summative evaluation techniques are two examples of commonly used usability tests.

2.5.1 Formative evaluations

A *formative evaluation* is a technique used in human-computer interaction and instructional design to audit a system while it is being developed to ensure that each component meets its principles objectives and to offer improvements during the construction process. If a system is designed to be used by a specific class of user, a representative member will typically be consulted as part of the formative evaluation process to ensure that the system developed is usable within the contexts of the target environment. An example of a formative evaluation technique is a heuristic evaluation, as discussed in §2.6.2.

2.5.2 Summative evaluations

In comparison to formative evaluation techniques, *summative evaluation* techniques are typically employed after the system has been designed and is used to ensure that the system, as a whole, meets the project objectives when used by members of the target group. Summative evaluations are often performed as user studies. User studies are discussed in §2.7.

2.6 Predictive evaluations

As usability testing can incur large costs, several *predictive evaluation* methods have been devised to lower the cost of evaluation by predicting usage rather than observing it directly. In most cases, the participation of end users is kept to a minimum as the costs associated with developing a testing plan and providing the facilities for these users may be difficult to justify, especially when a project is small in scale or when operating budgets are constrained.

2.6.1 Cognitive walkthroughs

Cognitive walkthroughs [24] involve the review of a system to uncover usability problems. These simulations are often conducted by experts of the system who emulate the behaviour of less experienced users to identify usability issues. Ideally, these reviewers would be experts

in the study of human-computer interaction and would have some knowledge of the target environment so that they can identify relevant problems rapidly and accurately. Using this technique, reviewers first determine the procedural knowledge necessary to perform the tasks required to satisfy specific objectives. This type of action mapping, best represented by the GOMS (Goals, Operations, Methods, and Selection Rules) cognitive model [14], consists of: a description of the steps necessary to perform a particular task; a list of the possible operations that can be actioned to accomplish each step; and a set of selection rules which determines the choice of action if multiple options exist that have the same end result. The reviewer then walks through the cognitive steps, making note of any mental obstacles or inconsistencies in the design that may prevent the user from completing the task.

Preece believes that the use of experts as “mock users” presents an advantage over using actual subjects because expert users tend to increase testing efficiency due to their familiarity with the testing process and are able to provide prescriptive feedback. “In terms of efficiency, a small number of reviewers can usually identify a whole range of potential problems for users during a single session. Real users would take much longer and require more facilities. Furthermore, experts are often forthcoming with prescriptive feedback about how the system can be improved and how usability problems can be put right” [36].

2.6.2 Heuristic evaluations

Nielsen popularized the concept of the heuristic evaluation, which was designed originally as a cost-effective solution for small companies that did not have the money or the facilities necessary to perform usability engineering [28]. In a heuristic evaluation, expert reviewers examine the system or prototype – in a fashion similar to a usage simulation – but guide their inspection based on a set of high-level heuristics that focus on specific types of usability concerns.

Selecting a set of meaningful heuristics for developing this focus can be difficult however, as the heuristics inevitably contain some overlapping and potentially contradictory advice [36]. In addition, the heuristics used to guide system development for a novice user may be quite different from the heuristics used to develop an optimized experience for expert users. Fortunately, many of these contradictions become less significant as the effects of higher-level principles and decisions diffuse into the lower levels of the design, but even so, inductive and deductive reasoning, combined with familiarity with the target environment, is critical to developing a pertinent set of heuristics.

Nielsen prefers heuristic evaluations [29], performed frequently on different facets of a system design, to large-scale usability testing as “elaborate usability tests are a waste of resources. The best results come from testing no more than five users and running as many small tests as you can afford.” The advantage of a heuristic evaluation is that it concentrates on a specific aspect of a system and, because of its relatively low cost, can be used frequently through the design process to verify the usability of the system as it is being developed.

2.7 User studies

A user study, also known as a usability test, typically involves the careful creation of a test scenario in which participants of the study are asked to perform a series of tasks while being watched by observers who take notes. User studies are generally summative in nature as they are run against well-developed systems for determining whether users of the system are able to complete the task for which the system was designed. In the event that the system does not meet its design expectations, the system would then be subjected to an iterative design cycle where solutions are found to rectify any usability deficiencies in the system before subsequent retesting. As user studies often involve subjects who are not active members of the research team, approval of the user study by an ethics committee is normally required to ensure: that the testing procedure is morally sound; that the anticipated cost of participation,

in time or inconvenience, is conveyed to the user; and that the proper releases are obtained from the subjects prior to the study's inception.

Chapter summary

In this chapter, we have introduced several of the essential concepts in the field of human-computer interaction and defined several techniques commonly used to evaluate system designs for usability. In the next chapter, we will discuss a specific type of user error that can be identified via usability testing: mode confusion.

3

Mode confusion

Having presented a general overview to the study of human-machine interaction in §2, we now narrow our focus on a specific type of error, *mode confusion*. In this chapter, we offer a detailed study of prior work on interface modes and explore some complementary research to provide a richer background upon which to discuss our own contributions. Of particular importance is a review of Degani's taxonomy of modes and an overview of interface interpretation errors as compiled by Leveson. By the conclusion of this chapter, the stage will be adequately set for the presentation of our own research in §4: a set of heuristics for the mitigation of mode confusion.

3.1 What is a mode?

Before our discussion about mode confusion and its intricacies, let us spend a moment to define what – in the context of this research – a *mode* entails. If a system (whether it be hardware, software, biological, or otherwise) is thought of as being composed of many disparate components, each with its own characteristics and identity, then a grouping of such, which expresses some underlying functional arrangement or condition, constitutes a mode. Leveson defines a mode as simply being a “mutually exclusive set of system behaviours” [23] but we expand upon this definition as modes are established with some organization in mind and are not the product of *ad-hoc* assignment. A mode does not need to contribute functional to the system, as being utile is only one of the criteria that can be used to organize a system, but a mode should yield some sort of organizational benefit in a coherent design.

Colour, for example, may not have utility as a variable in itself, but can be used nevertheless to categorize. Akin to this notion of functional composition, several lower-level modes can likewise be condensed together, yielding higher-level modes that, when further aggregated, ultimately describe the entire feature set of the system.

The exact scheme used to arrange components into modes varies with each design, but in most cases, systems are partitioned into subsets based on effect — the benefit of which depends on what aspect of a system the designer favours. For example, a programmer can advocate an organization scheme that favours the initial creation and subsequent maintenance of the system at the expense of run-time efficiency. Alternatively, the designer can partition a system to optimize the speed of the system even if it means a reduction in the ease of maintenance. A blend of both strategies typically offers the best balance, but the important realization is that any logical organization is inherently biased and will emphasize the importance of one attribute or set of attributes while simultaneously diminishing others. In practice, an organization scheme that may be good for the programmer oftentimes incurs an operational penalty for the end-user, while developing a user-centric design typically forces the development of *middleware* – a layer of abstraction that does not contribute to the functionality of the system, but is essential to the human-machine interface. Different biases can be used at different levels of a system, with modes favouring the technical aspects of a system at the low-lying levels, and modes that favour usability at higher levels.

3.2 Why do we use modes?

As partitions in logic, modes are used to manipulate the flow of information and control in a system by dividing systems, so that they are more manageable for the system designer and the user, or restricting the number of choices afforded by the system's internal mechanisms to the user in a given scenario. The usefulness of modes should be evident following the next section, an overview of the different types of modes as classified by Degani.

3.3 What are the different types of modes?

Degani has classified modes into three distinct variants: *interface modes*, *functional modes*, and *supervisory modes* [7]. Interface modes are modes that exist at the presentation level of a system (i.e. the user interface), functional modes regulate a system's internal mechanisms, and supervisory modes control the balance between autonomy and interaction in an automated system.

3.3.1 Interface modes

Interface modes are ordering mechanisms that control the appearance or flow of control in mutable user interfaces. Applied to the presentation layer, interface modes modify the behaviour of the user interface in order to improve the efficiency of the human-machine interface. Interface modes not directly bound to the underlying system mechanisms, however, as they do not control the behaviour of the system's internals but are restricted to customizing the human-user interface for best efficiency. A system engineer charged with the development of an interface mode would benefit from a review of basic human factors principles, as the differences between a well-designed interface mode and a poorly conceived one that reduces efficiency are subtle but vitally important.

In their most popular manifestation, interface modes are used to contract or expand the size of the interface's input or output space, with the hope of reducing the cognitive load on the user by limiting the number of options he or she needs to evaluate before being able to choose an appropriate response [37]. This technique can be thought of as being similar to *information hiding* except that it is not data that is being hidden from the user but rather choices (Figure 3.3.1a). The most important side-effect of this *channelling* process is that the options which have been pre-eliminated by the system designer are typically undesirable options and, by extension, would yield undesirable system behaviour if activated given the current context. For novice users, this process is extremely beneficial as it acts as an assistive

exploratory guide so that the user – who may only have a superficial understanding of the system – can find the option they want and completely circumnavigate those less desirable options that could be inappropriate or dangerous.

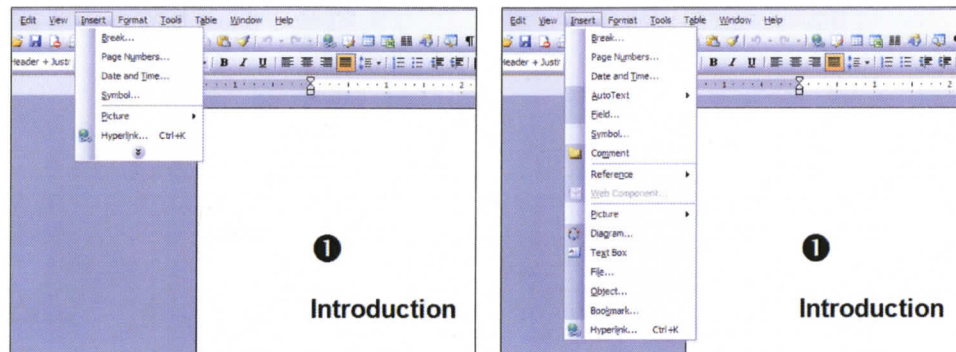


Figure 3.3.1a: Later versions of Microsoft Word include an adaptive user interface to show only essential and recently used menu options (L). To see the hidden options (R), the user can either expand the menu or disable the adaptive behaviour completely.

Unfortunately, this type of *restricted information presentation* can produce users who have only a limited understanding of the system, a flawed understanding of the system, or no real understanding at all since the choices given to the user are restricted at each turn. Unless a user has the opportunity to exhaustively traverse each branch of the decision tree and map out the various options, they may never be able to gain a complete understanding of the system, a prospect that could prove dangerous in the event of non-nominal system behaviour. *Expert systems* and *wizards* are two examples of step-based procedures that use restrictive information presentation techniques.

Another common interface mode is the *restricted information input*, in which a system designer pre-defines the type of data allowable as input. As computer systems are every poor at dealing with anomalies, it is important that the data fed to the system is meaningful and is in the format or of the type expected by the system. In programming languages, this type of input restriction is known as *type checking*, whereby the structure of the data is matched with

the documented inputs for a function to ensure congruency. As an interface modification technique, restrictive input prevents a user from entering unanticipated values that could lead to errors in processing (Figure 3.3.1b).

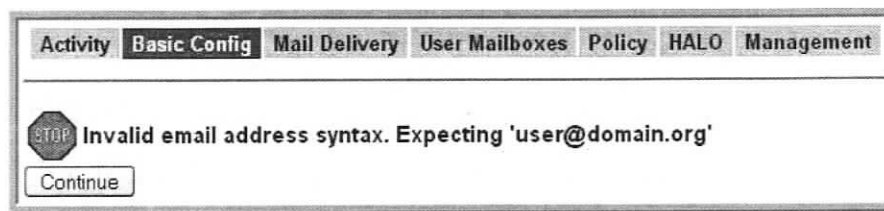


Figure 3.3.1b: An example of type checking: this application is expecting an email address in the standard "user@domain" format as defined in IETF RFC 821.

Rudimentary type testing is unable to stop the user from entering syntactically correct, but semantically invalid values however – such as the entering a North American telephone number beginning with a zero – so more complex protective mechanisms are necessary to explicitly forbid certain values; the actual values excluded would depend on the particular system context. Semantic checkers are complicated to develop and use, however, as the system would need to be provided intelligence capable of detecting semantically a flawed input that could yield undesirable behaviour if processed.

A third and fourth type of interface mode are menu-driven systems and expanding hierarchical trees. In the former, the juxtaposition of related options are aggregated together into groups (Figure 3.3.1c), while in the latter components are placed in layers such that *providers* (parents) of dependencies are placed at higher strata relative to the *dependents* (children) which are situated on lower strata. In both cases, the bulk of the system's data and choices are initially hidden from view and portions relevant to the current operation are presented on demand as the user's query is further focused through the traversal of the menu system or hierarchical tree.

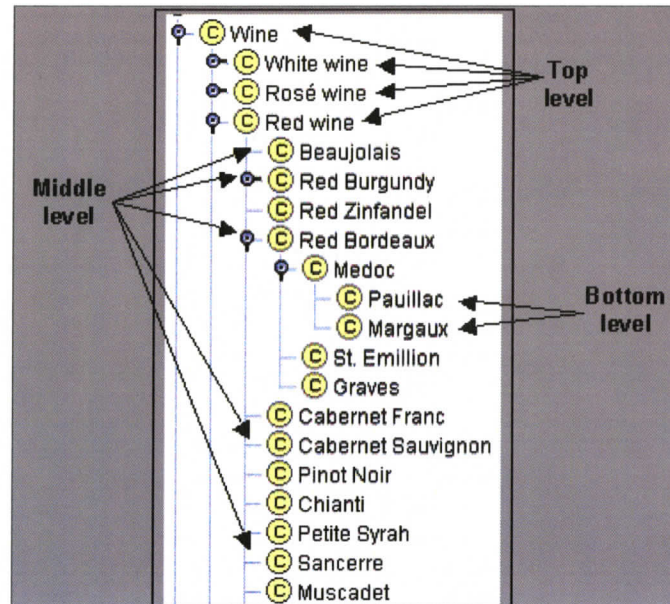


Figure 3.3.1c: An expanding tree, with options grouped based on some logical hierarchy, allow designers to hide information at different levels, revealing only parts of the information as requested by the user. This scheme reduces clutter and reduces the mental workload of the user.

The effectiveness of such system designs are directly attributed to the system designers who are tasked with the development of semi-intelligent, context-aware algorithms that govern the visibility of the user interface components. In almost all cases, the patterns used to develop this revealing behaviour are derived from use cases. *Use cases* are scenarios of usage predetermined by the system designer as potential ways of utilizing the system's affordances and resources to produce a specific effect. They are task-based in nature and normally describe a process starting from a known system state to a point where the system yields the desired result. While an exhaustive search for all possible uses of a system would provide the greatest amount of coverage, a review to such an extent could be prohibitively expensive. Even if the cost was ignored, such a search could still prove futile, as it is impossible to prescribe all possible situations in a system where unbounded variables exist. As such, use cases are weighted based on frequency of use and importance of function, with only the most heavily weighted considered when formulating an interface mode's visibility behaviour. In a

similar vein, use case weighting can also be used by a system designer to determine the ordinal sequence in which options are presented to the user, in menu-driven systems and hierarchical trees, when more than one equally important option is presented at the same strata.

3.3.2 Functional modes

Whereas interface modes control the flow of information to and from a user, functional modes govern the flow of information input and output within a system. Functional modes are relegated to the internal mechanisms of the automation and typically are not exposed to the user, except in the case of *functional mode annunciators*, which are indicators that expressively tell the user of the internal functional mode of the system (see §4.2: heuristic B1). More commonly, the effects of functional modes manifest themselves indirectly in the user interface by way of an interface mode that has its visibility behaviour tied to an underlying system mechanism.

Functional modes are commonly used in two fashions: to enforce limitations in system behaviour for the purposes of user safety and protection against errant behaviour; and to partition the system's behaviour into manageable subsets for the ease of development and subsequent management by the system designer. In the first case, functional modes can be used to enable and disable portions of the system to prevent the automation from engaging in undesirable behaviour or falling into a hazardous state. These limiting factors, like those that govern visibility behaviour, are derived from the use cases that are determined *a-priori* by the system designer. However, as it is difficult to determine all possible use cases of a system, it is likewise daunting to describe all possible factors that could lead to detrimental system behaviour. As a result, the effectiveness of functional modes to constrain errant behaviour is limited by our general inability to describe all possibilities. For simple systems with a small number of variables, it may be possible to construct a set of rules that could provide adequate

coverage of all foreseeable circumstances but in large systems with thousands of variables, it may well be impossible to guard against more than the most critical of scenarios even though a larger body of system failures is known. There has been much research on the topic of *critical states* and *hazardous states*, especially in the context of software engineering and safety-critical systems; consult Neogi's [27] work for a thorough discussion of critical and hazardous states.

The second common use for functional modes is to decompose a complex system into its elemental components so that they can be more easily handled by the system designer for development or maintenance purposes. This arrangement is for the mental benefit of the programmer and should have no bearing on the operation of the system as, with respect to the system specifications, it does not matter whether a system is divided into ten, one hundred, or one thousand constituent parts (or along which lines these divisions occur). When functional modes are used to dissect a system, they behave very much like interface modes in that they allow for information or functionality in the system to be hidden or ignored to help focusing the attention of the programmer on a specific facet of the system.

3.3.3 Supervisory modes

The final category of Degani's modes, the *supervisory mode*, governs the level of interaction between the user and the system. Supervisory modes are frequently used in machines with some measure of autonomous operation; where a system gathers inputs, makes a logical decision based on its programming, and commits to a corresponding action, all without consultation of the user. In such circumstances, the user's role is reduced from actor to bystander, as their function is now to oversee the process and ensure that the automation behaves as expected. This delegation of responsibility is particularly contentious amongst designers as research [42] has shown that once humans are "taken out of the loop", their ability to monitor and intervene can become severely impaired.

One common example of a supervisory mode, typical in mechanical automation, is the *manual override*. When a manual override is activated, the autonomic controls of a system are disengaged and the user is given full responsibility over the analysis of data, formulation of appropriate actions, and commitment to any subsequent action. In a simple, step-based system where synchronicity and chronometry is not a concern, manual overrides may quite successful but in high-complexity systems, the term “manual override” is really a misnomer as the processes in a system are typically too complex and delicate to survive the imprecise and awkward manipulation of humans. In such cases, “manual override” really suggests an *emergency stop* – the immediate halt of system operation – rather than a wholesale transfer of control into the hands of the user. Once the reason for an emergency stop is addressed, the system is usually put back into a known *safe state* before control is relinquished to the automation.

Another common manifestation of a supervisory mode is the *confirmation dialogue*. Confirmation dialogues are scattered liberally throughout most user interfaces but are used largely for two purposes: to ask for the user’s accordance before performing an action initiated by the system; and to double-check a user’s intention before performing a potentially dangerous action. In the first case, the system designer has determined that it would be prudent for the system to prompt the user for concordance before actioning for either informative reasons – to grant the user a better understanding of the system’s operating behaviour through increased situational awareness – or because the action about to be performed is particularly dangerous if activated at the wrong time (see §2.3 regarding errors of commission). This first case is a sanity check for the system’s logic while the second case is a sanity check for the user’s action as the confirmation ensures that the user has acted with premeditation.

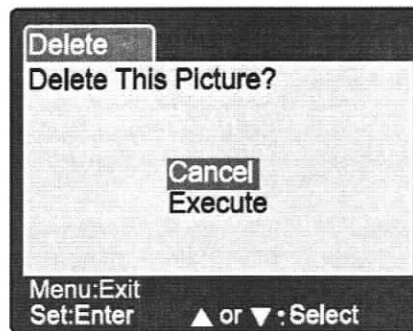


Figure 3.3.3: A confirmation dialogue is used in the Gateway DC-T50 digital camera as a flow block to prevent the accidental deletion of images.

In both cases, the system designer has determined ahead of time that the request is a significant action that should not be entered into lightly, and the confirmation dialogue acts as a *flow block*; a mechanism intended to disrupt the mental flow of the user by introducing an additional task to force further confirmation of intention before action (Figure 3.3.3). Flow blocks can lose their effectiveness overtime however as increased familiarity with the system and anticipation of this resistance, coupled with the user's mounting muscle memory, can create a situation wherein the acceptance of the confirmation is automatic in nature and done without any thought on the part of the user. This type of *pre-emptive acceptance* renders the confirmation dialogue useless; a possible solution to counteract this type of anticipated resistance is described in §4.2: heuristic C2.

3.4 What is mode confusion?

In a system where there are more functions available to the user than the number of controls for manipulation, some or all of these controls must take on multiple roles, changing personality based on the system context. Similarly, when the number of possible system states is greater than the number of available indicators, annunciators are overloaded to convey different meanings under different circumstances. Mode confusion is the term given to the scenario in which a user finds it difficult or impossible to determine the current state of

the automation due to *errors in interface interpretation* (§3.7.1), *inconsistent system behaviour* (§3.7.2), *indirect mode changes* (§3.7.3), *unclear operator authority limits* (§3.7.4), *unintended side effects* (§3.7.5), or *lack of appropriate feedback* (§3.7.6) and as a result performs an *error of commission* or an *error of omission* (§2.3).

3.5 Why is the study of mode confusion important?

The need to devise techniques to alleviate mode confusion is becoming increasingly important as more automated systems are making use of modes. Market pressures demand that each successive generation of product represents an increase in functionality and simultaneous decrease in physical size (Figure 3.5). In what Sarter, Woods, and Billings calls *technology-centered automation* [43], technology takes centre stage and operators are taught to accommodate the system's shortcomings because it is easier to train a user to fit the system than to build a system to fit the user.

The current trend in technology design is to grant automation increasing amounts of responsibility and independence, shifting human operators to largely supervisory roles. By becoming the principle actor in the system, the automation has reduced demands on the operator in a way, but by distancing the user from the actual process, users may lose familiarity with how the system operates, possibly undercutting the operator's ability to be effective should the automation fail. Compounded with the need for a user to maintain an accurate mental model to anticipate and predict future system behaviour – while attempting to catch potential system malfunctions – and it is clear that the demands on the user, from a human-machine interface perspective, have in fact increased.

Alleviation of mode confusion has the potential to reduce significantly the cognitive workload of users by minimizing or completely eliminating efficiency-robbing interface interpretation errors.



Figure 3.5: The mobile telephone has been the subject of exceptional advancement over the past twenty years. The first cellular telephone, Motorola's DynaTAC 8000X (L), weighed 795 grams and lasted 30 minutes per charge. That was 1984. Motorola's latest mobile, the RAZR V3 (R), tips the scales at a svelte 95 grams and yields up to seven hours of talk time. (Images are not to scale.)

3.6 Modes and mental models

Unlike the system designer who is intimately acquainted with the internal mechanisms of a system, the user's understanding of a system is based solely on the user's training and any insight gleaned from first-hand experience using the system. The user's actual level of comprehension may be several degrees less than the programmer's and yet still be able to use the system with some efficacy provided they are able to develop a *mental model* of the system's functionality that is tolerably congruent with the system's actual operating model. Mental models are internalized conceptions of how a system operates and are developed through explicit training or self-discovery [31].

Cognitive scientists have proposed that humans construct mental models of reality to anticipate and to reason. The theory of mental models is founded on three core principles: that each mental model represents a relationship between two entities, physical or ethereal; that mental models represent what is true according to premises; and that deductive reasoning relies on mental models [30]. Johnson-Laird [15] believes that once a framework for a mental model is established, connections representing all possible relationships between the two entities will be made, even though many will be subsequently refuted through experience. Johnson-Laird's principle of truth further postulates that humans, by nature, do not record false connections in their mental models but rather make "mental footnotes" about the falsity of certain relationships and that a user's mental model therefore includes not only what is correct but will, by way of understanding, also include knowledge of what is incorrect.

In the formative stages of comprehension, this type of *ad-hoc* learning is particularly important as it allows users to internalize a semblance of the underlying system with a provision for further refinement over time. The benefit of having such a mental model is that a user can then execute "what-if" evaluations internally, without the need to acquire empirical data, that could lead to further understanding of the system and an ability to anticipate real-world behaviour. These mental models are automatically built-up over time and with each successive interaction, the model is updated so that it best reflects the latest user experience; correct assumptions are positively reinforced while contradictory assertions are demoted or eliminated from the mental model.

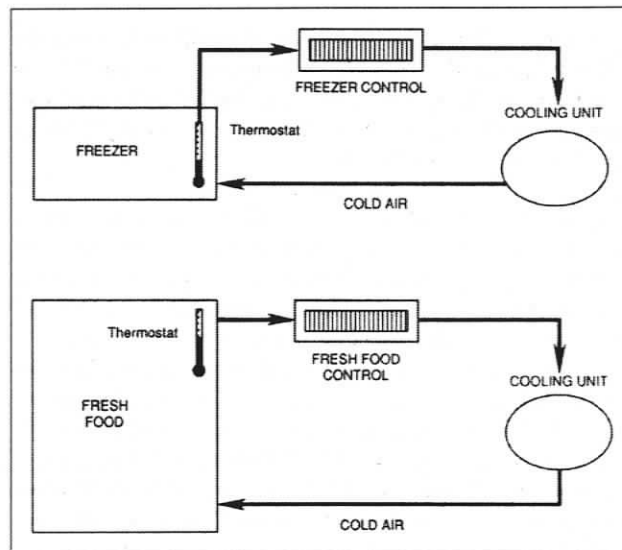


Figure 3.6: Norman's mental model of his refrigerator, as derived from the documentation provided by the manufacturer and examination of the various controls. Compare this to Figure 3.7.5, which is the actual system model as built by the refrigerator's engineers.

The accuracy of documentation supplied to the user is also important to the success of a product, particularly if the design includes many difficult to use functions or difficult to understand concepts. Poorly written documentation may foster the development of incorrect mental models through the over-simplification of technical details and may be more detrimental to the usability of a device than insufficient documentation. Insufficient documentation will leave the user with unanswered questions which will naturally arouse caution in the user, while inaccurate documentation may provide the user with a false sense of certainty because documentation is typically viewed as being authoritative and truthful.

With enough time, the user's internal model will reach a level of sophistication and refinement that the user may only experience positive reinforcement and have no counter indications. The user may then confidently believe that their mental model is an accurate representation of the underlying system mechanisms even though their understanding may actually be flawed (Figure 3.6). What happens when the behaviour of the system suddenly

differs from the user's expectations? Does this inconsistency imply that the user's mental model is faulty, or does it mean that the system's mechanisms have failed in some way? Moreover, if the user is not supercilious enough to believe their own perceptions are infallible, can the remaining vestiges of the flawed mental model be used to rectify the situation?

3.7 Why do good mental models go bad?

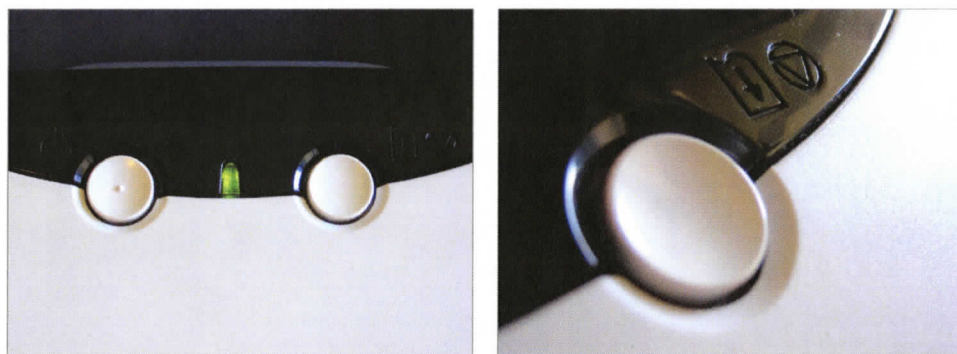
Leveson believes that *interface errors* are caused largely by a divergence between a user's mental model and the actual system model [38] and criticizes user interfaces for providing infrequent feedback or feedback that is contradictory to the system model. Regular and elucidating notifications, Leveson contends, is vital to the proper development of a mental model and can stamp out erroneous assumptions about a system early in the model's development. Several sources of contamination, collectively known as interface errors, have been identified by both Leveson and Degani: *errors in interface interpretation, inconsistent system behaviour, indirect mode changes, unclear operator authority limits, unintended side effects, and lack of appropriate feedback.*

3.7.1 Errors in interface interpretation

Errors in interface interpretation are the dissonance that occurs when a user's output is incongruent with the system's expected input. The root of such errors is almost always ambiguity wherein the user is unable to properly ascertain the nature of the input requested by the system and as a consequence provides the automation with an input that is inappropriate (see §2.3, regarding errors of commission). Sometimes, the input may simply be formatted incorrectly, while at other times the user may have misinterpreted the system's query and provided an answer that is invalid given the current system context but may be appropriate in a semantically or syntactically similar context. With respect to culpability, many users mistakenly blame themselves for this incongruence but Norman believes that the

fault lies squarely in the design of the user interface as all interface components (i.e. widgets, labels, and annunciations) should be designed with lucidity and without the possibility for misrepresentation, especially in systems where safety is a primary concern [20].

One specific example of this ambiguity is *annunciation duality*. For cost reduction and form factor reasons, many binary annunciators (indicators that can be either on or off) are *overloaded* to represent more than one system condition. When binary indicators are used to indicate a nominal overall status, this type of design ambiguity is acceptable because the system does not require user intervention. In cases where multiple error conditions are annunciated using a common indicator however, the user is left with an inability determine which component of the system has erred as no specifics are being conveyed (Figure 3.7.1a). The usefulness of such an error indicator is greatly lessened as it announces that some exceptional condition has occurred but does not communicate the source of the error. As a result, the user is left with no recourse but to follow a potentially exhausting and technical troubleshooting checklist in order to determine the cause of the fault and, unless the first item on the checklist is identified as the cause, many faultless components of the system may need to be checked in sequence before the cause can be isolated and rectified.



3.7.1a: Many systems, like this inexpensive inkjet printer, use a minimal set of generic annunciators to indicate multiple system conditions. The single annunciator (L) has four states: off, slow blinking, fast blinking, and steady on. Unfortunately, the printer markings (R) offer little assistance in mapping the annunciator's states to system conditions.

Another potential interface interpretation error is more subtle but equally problematic. In designs where multiple modes of operation share a common textual or numeric output (Figure 3.7.1b), the difference between the annunciations for each mode may be as subtle as a single inconspicuous indicator – such as a decimal point or difference in colour or gradiation – which an uninitiated user may dismiss as simply a superfluous anomaly or stylistic adornment.

Degani identifies a third source of interface interpretation errors, *circular mode transitions*, which are in essence a form of annunciation duality but in reverse. In annunciation duality, two inputs are mapped to a common output while in a circular mode transition a single input is used to trigger multiple and successive mode changes within the system. Without proper indication that such a mode change has occurred, the user can become quickly disoriented as they lose contextual awareness of the underlying automation's state. If this change in mode goes unannounced, unnoticed, and uncorrected, the user may inadvertently enter values corresponding to another mode of the system, potentially leading to a system error or hazardous condition. Errors in interface interpretation are readily preventable through the employ of user-centric interface design sensibilities.



Figure 3.7.1b: Digital alarm clocks commonly use a single indicator to differentiate between the interface modes for (L) time setting and (R) alarm setting.

3.7.2 Inconsistent system behaviour

Another common source of mental model incongruence is inconsistent system behaviour. Because a user is more likely to be familiar with the intimate details of a frequently used system component, the mental models developed in these established portions of the system are often called upon when unfamiliar, but logically similar circumstances present themselves [15]. If an esoteric component's behaviour is in-line with known parts of the system, the user stands a good chance of being able to successfully *ad-lib* the interaction sequence and produce an acceptable result. If, however, the system fails to respond as expected, then even a seasoned user may have no better inclination as to how to bring about a successful resolution as an inexperienced user [32].

A design is consistent when a similar task or goal is associated with a similar or identical action [9]. Uniform behaviour allows a user to leverage knowledge stored in an established mental model to accommodate unfamiliar tasks, which is particularly important in fast-paced or high-stress environments. If small differences are detected, the user may still be able to use an existing mental model if only slight adaptations are required to accommodate the new situation. In extreme cases where the inconsistency is significant, however, the user may be forced to completely abandon any cached information – even if parts of the mental model are still serviceable – and begin the learning process afresh from a rudimentary level. Regardless of whether the effort required to adapt is slight or significant, the existence of even the smallest monad of inconsistency robs the human-machine interface of efficiency and places pressure on the individual to adapt to a new pattern of behaviour in a timely manner. The amount of pressure exerted is directly related to the timeliness of the action being commissioned and in industrial production environments, or when working with transportation automation, the time available to an operator for such intensive contemplation and *ad-hoc* learning may be minutes if not seconds.

3.7.3 Indirect mode changes

Indirect mode changes are the most insidious of the interface interpretation errors discussed by Degani and Leveson because they occur without any action on the part of the user. Such operational mode changes are triggered by pre-defined conditions embedded in the automation logic. If an indirect mode change occurs commonly and predictably during system operation, a user will incorporate this into their mental model and build up an expectation that an uncommanded mode change will occur. Unfortunately, most indirect mode changes take place during periods of atypical behaviour as they are triggered by system protection mechanisms that are designed to keep the automation from entering a potentially dangerous mode of operation. Ironically, users who are unaware of such fail-safes may falsely believe that the system is malfunctioning, due to its seemingly errant behaviour, and attempt to compensate for the uncommanded output by applying an equal but opposite input. This concept of “overriding an override” is discussed in the next subsection: *unclear operator authority limits*.



Figure 3.7.3: Electric motors in light-duty appliances, such as hand blenders, hair dryers, and electric drills, and are not designed for continuous operation and typically require a period of rest after prolonged use. For this (L) paper shredder, a 30-minute cool down period is required after two minutes of continuous operation (R). Users unaware of this behaviour may think that the shredder has malfunctioned.

In the other types of interface errors described, the user has performed an action and is expecting some sort of feedback, positive or negative, from the system. Indirect mode changes are problematic because they are uncommanded and may go completely unnoticed, as the user is not expecting any feedback from the system. Circular mode transitions differ from indirect mode transitions in that the former is at the request of the user while the latter, is an uncommanded mode change. Sarter and Woods believe that the most common type of error to come about from indirect mode changes are errors of omission [41], which were discussed previously in §2.3.

3.7.4 Unclear operator authority limits

In systems where there exists the potential for dangerous behaviour, *operator authority limits* are commonly employed for safety and can take the form of an interlock or lockout. *Interlocks* are step-based process controls commonly used to enforce the correct sequencing of actions or to isolate two events in time and can be implemented as a mechanical lockout – where some physical impediment is introduced which precludes unsynchronized operation – or incorporated into the system’s programming such that specific conditions must be satisfied before subsequent action is allowed. A *lockout*, in comparison, is a more advanced mechanism that not only makes it impossible or very difficult for a system to enter a known hazardous state, but will assume control of a system to prevent this hazardous condition from occurring, overriding a user’s inputs if necessary. Because this supplantation reduces or completely removes the ability of the user to affect change, it is used infrequently and only in cases where the potential for harm is great (Figure 3.7.4). Notification to the user that a reduced or lack of control should be expected is vitally important to forestalling an interface interpretation error.

The circumvention of a lockout is never exercised except in extraordinary conditions and is never done purposefully as part of a nominal use case; confirmations are designed to hinder

but not inhibit while lockouts are designed to inhibit authoritatively. Users who are unfamiliar with the system's autonomous protection methods or are unaware that such protection exists may struggle against the system; a conflict made even more serious as it is typically only during periods of high-criticality – when user attention is already at highest and the consequence of failure the most severe – that protection mechanisms are employed.

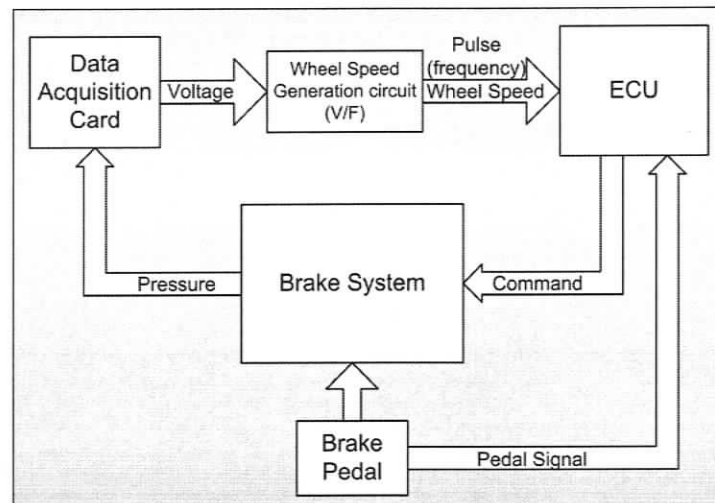


Figure 3.7.4: An automobile anti-lock braking system ensures that the each wheel remains rotating during heavy braking, regardless of the amount of pedal pressure supplied by the driver. When the ABS system is activated, brake force is determined by the automation, not the driver.

3.7.5 Unintended side effects

A unique case of system ambiguity, *unintended side effects* occur when the intentional initiation of an action not only results in the desired effect but also produces a secondary unexpected action. These unintended side effects are often the result of programming errors and are rarely introduced on purpose; a better alternative would be to use a deliberate mode transition with proper user notification. Unintended side effects can lead to interface interpretation errors in similar to indirect mode changes as the system automation has

deviated from the expected state envisioned by the user's mental model, nullifying the user's ability to accurately predict and anticipate subsequent system behaviour.

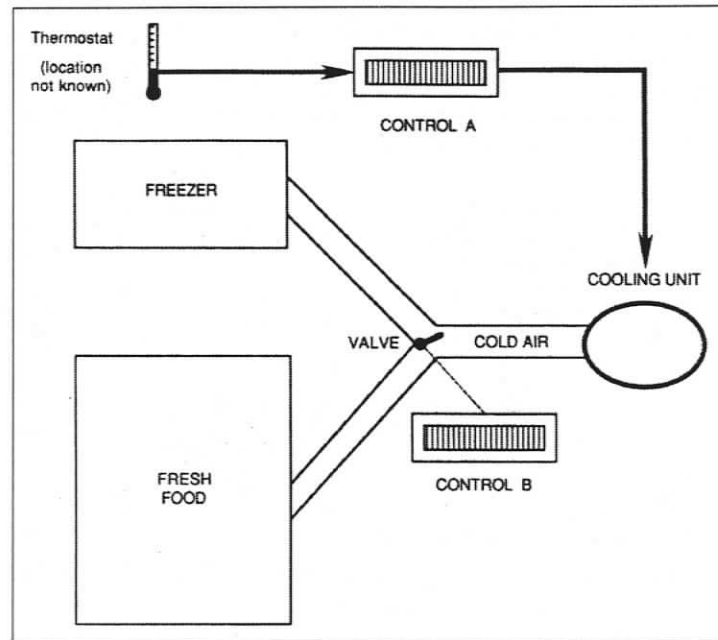


Figure 3.7.5: In Norman's refrigerator, turning off the freezer compartment also produces the unintended side effect of turning off the chilling compartment. This behaviour is not noted in the manufacturer's documentation but is obvious from this diagram of the system model. Compare this Figure 3.6.

Leveson is quick to point out, however, that unintended side effects differ fundamentally from indirect mode changes as side effects need not be mode changes in themselves but can manifest themselves as modifications of other system variables used to determine behaviour (such as data values held in computer memory) [23].

3.7.6 Lack of appropriate feedback

In order to determine if a system's operation is congruent with an internal mental model, the user must be provided with accurate and timely information pertaining to the current

operational state of the system. If a mismatch between the observed operation and the anticipated operation is detected, the user must then determine whether it is the system's operation that is at fault or if the user's mental model is deficient and needs amendment. Feedback can be obtained through provisions incorporated into the system design or by user observation through the direct monitoring of system mechanisms or by recognizing that a known side effect, external to the process under scrutiny, has occurred. This status information is vitally important to the "feedback control loop" [2] as a user who is not able to determine when a mode change has occurred will be unable to keep operational synchronicity between the system model and their mental model.

Designing a system to provide an appropriate level of feedback can be challenging as care must be taken not to burden the user with too much information. This situation, termed *information overload*, can be as detrimental to a user's ability to analyze a situation as the case where too little information is provided [45]. In both regards, the user is unable to find the information they need in a concise manner, making it difficult to render an assessment of the system's health. A balance between visibility and invisibility must therefore be established that highlights the salient features of a system's operation while keeping strictly internalized variables, such as those that have no affect on overall system operation or mode transitions, from public exposure.

Finally, the quality of the information conveyed during feedback can also suffer degradation as the health monitoring and trouble diagnosis mechanisms are provided by subsystems that are themselves automated and therefore prone to failure. Just as a user will compare his or her mental model with the expected system behaviour, the automated error detection devices also contain their own system models, usually implemented as a series of logical statements, to detect errors. Unfortunately, the quality of error detection is only as good as the design of the monitoring system and validity of the data arriving from the sensors, so appropriate

feedback may not be generated if the system is confused about the operating environment or its own internal state [38]. This adds an extra level of complexity to the system as annunciated errors may refer to errors in the core system or in the monitoring subsystem itself, necessitating a third-party diagnosis.

3.8 Requirements engineering and mode confusion

Much of Leveson's work in mode confusion is strongly associated with requirements engineering [23]. Requirements engineering is the branch of engineering research dedicated to developing techniques to optimize the process used to develop and specify a product's functional requirements. The proper specification of requirements for a system, often derived from the intended usage scenarios for the system under design, is particularly important as the requirements directly influence the uses afforded by the system. If there is a disjunction between the intended use of the system, as codified by the engineer, and the perceived uses of a system, as interpreted by the user, then there is a large potential for mode confusion. As mentioned in §1.1, digital cameras are an emerging technology that is experiencing continual refinement and redefinition and as such, there is a need for manufacturers to ensure that their requirements specifications evolve over time to properly reflect the changing expectations of digital camera users.

Chapter summary

In this chapter, we discussed how interface interpretation errors can lead to *mode confusion*, a situation where a user is unable to determine with accuracy the underlying state of an automated system. The chapter began with a definition of *mode*, listed several types of modes as categorized by Degani, and continued with a definition of *mode confusion* and a discussion of why we should be concerned about the proliferation of modes in today's automated systems. Our discussion then moved to the work of Johnson-Laird who postulated that humans develop internal representations of system behaviour known as *mental models*,

and finished with a collection of interface interpretation errors, identified by Leveson, that Norman believes are a result of incongruence between a user's mental model and the system's actual model.

In the next chapter, we discuss the main contribution of our research, a set of heuristics that has the potential to mitigate many common interface interpretation errors from user interface designs, reducing the likelihood of mode confusion.

4

A set of heuristics for mitigating mode confusion

While §2 provided an overview of the theories of human computer interaction with an emphasis on those theories pertaining to user interface design, §3 focused on a specific type of error common in user interfaces – interface interpretation errors – and detailed some of the factors that can contribute to mode confusion. In this chapter, we develop a set of heuristics to mitigate mode confusion that leverages the theories of §2 to avoid the errors of §3.

4.1 Why is it important to develop these heuristics?

As discussed in the previous chapter, mode confusion is a serious flaw of contemporary user interface designs [25]. In its most benign form, mode confusion can be an annoyance that causes the user to make inconsequential slips, resulting in a reduction of efficiency, but when mode confusion occurs in mission-critical systems, errant behaviour can have dire consequences [12]. The ability to alleviate mode confusion is a desirable and worthwhile avenue of research [22].

Before we begin formulating a set of heuristics for controlling mode confusion, we must first acknowledge that mode confusion is not an intrinsic aspect of user interfaces design and that it can be separated from the core functionality of the system. If such disassociation cannot be accomplished, then the system may need significant reworking if mode confusion is to be

reduced. Note that the complete elimination of mode confusion may not be desirable as the need to follow per-established conventions may outweigh the need for absolute correctness.



Figure 4.1: An example of social convention and aesthetics overruling usability. The design of a typical entry way door handle affords two actions: pushing down on the button and pulling. Unfortunately, the front door of most homes always open inward and the ergonomics of the handle makes pushing awkward.

From the user errors discussed in §2 and the interpretation errors of §3, a set of heuristics have been designed to help engineers identify situations with mode confusion potential in their interface designs. These heuristics are meant to invoke thought in the programmers, with respect to human-computer interaction concerns, and are designed to challenge the assumptions that designers may make about their potential target audience. These heuristics should be considered early in the development cycle, possibly as a formative evaluation technique, for maximum impact as preliminary system decisions can significantly colour future usability characteristics. Indeed, these heuristics can also be considered during the requirements gathering stage of a product's definition to ensure that the mitigation of usability issues in general, and specifically, mode confusion, are identified in the development process as a component integral to a successful design. Nielsen believes that

when heuristic evaluations are done in groups, as little as five evaluators can find roughly 75% of the usability problems in a system [28]. Performed at an early stage, heuristic evaluations can address design problems before they become entrenched in the system as usability issues in the core of a system can be both costly and time consuming to resolve. These heuristics are not exhaustive however and should not be implemented as a set of prescriptive design guidelines.

The reasons for this limitation are three-fold. First, consider the scenario wherein these heuristics are used as dictatorial specifications on which a system is built. As human interaction theory and mode confusion mitigation research are not holistic system engineering disciplines, there is no guarantee that a system built using this foundation would be able to produce a viable system design. Depending on the nature of the system required, these heuristics may be wholly incompatible with the actual structure necessary, preventing the successful completion of the system because usability characteristics and interface efficiency targets were enforced.

Second, as it is the goal of these heuristics to reduce or possibly eliminate potentially confusing design elements from a system, using these heuristics as concrete restrictions may cause designers to become complacent in their designs as they may erroneously believe that they have solved every usability problem in the system once these heuristics have been implemented. The limitations of this research must be considered with due gravity because these heuristics are not exhaustive and subsequent designs can introduce new types of flaws that this technique does not encapsulate.

Finally, an explicit use of these heuristics as assertions could potentially stifle the advancement of user interface designs by imparting a false sense of finality to user-interface design research. As it is easier to be critical than to be correct, it is our aspiration that these

heuristics will foster the continued development of innovative solutions to user interface design issues. As the heuristics are not mutually exclusive of one another, and will inevitably contradict one another, trade-offs must also be made between competing heuristics to distil a subset of heuristics that best match the specifications required to meet the system objectives.

It should be noted that the body of knowledge contained within these heuristics are not entirely novel as they have their roots firmly planted in the established theories of human-computer interaction and human-factors engineering. What is unique is that these recommendations have been expressed as a set of heuristics that will allow designers – whether they have a strong background in human factors or little or no interface design experience – to identify situations with mode confusion potential. A heuristic evaluation and user study was performed to determine the usefulness of these heuristics; see §5 for a description of the heuristic evaluation process and §6 for a discussion of user study.

4.2 For the designer: A note before using these heuristics

This section and §4.3 have been written for easy reading by system designers and as such may be somewhat brief and use terminology that is common vernacular for the target audience. The change in writing style and tone is intentional. Where appropriate, applicability recommendations and illustrations have been provided alongside each heuristic to indicate where a certain technique could be of particular benefit and similarly warn against the potential pitfalls of misapplication. References to other sections of this thesis are shown in italics and are for the benefit of the reader; as a deliverable to the engineer, these references would be omitted.

What is a heuristic evaluation?

Heuristics allows for the identification of usability problems through introspective evaluation. Heuristics can be performed on a system at any stage of development, whether it is finished,

under construction, prototyped, or only a mental concept. *For additional information about heuristic evaluations, see §2.6.2.*

Why should I use these heuristics?

Heuristics are especially effective when used early in the development cycle as potential usability problems can be found and resolved before a single line of code has been written or any effort has been invested in constructing a prototype [29]. The time and money saved by solving a problem – before it becomes a problem – should not be underestimated. Studies have shown that when heuristic evaluations are done in groups, as little as five evaluators can find roughly 75% of the usability problems in a system [28].

How is this technique different from Nielsen's usability heuristics?

These heuristics are derived from the knowledge of many sources – including Nielsen's [26]. What makes this heuristic technique different is that it focuses specifically on the problem of mode confusion. *For a discussion of why alleviating mode confusion is important, see §3.5.* As these heuristics are tailored to catch mode confusion problems, Nielsen's heuristics can be used in parallel for a more complete coverage of general usability issues.

Should I use these heuristics instead of performing a user study?

Heuristic evaluations and user studies are complementary techniques for usability testing that are frequently employed in tandem [9]. Each methodology has its advantages and disadvantages and are often used at different phases in a system's development — a heuristic evaluation is often performed early in the design phase before the system has been fully defined and can help guide the development process, while a user study is typically used later in the design process after a mock-up or prototype has been built [36] and has the advantage of providing detailed empirical results [1]. *For a discussion of formative and summative evaluations, see §2.5.1 and §2.5.2.*

How should I use these heuristics?

Below are 14 heuristics that are designed to make you think about your design – with emphasis on the user’s point of view – and to challenge any assumptions you may be making about your target audience. Consider each heuristic in turn, evaluating your system against these generalized rules-of-thumb, to identify potential usability problems. If you are working with others, collate the lists of identified issues and decide as a team which problems are the most important and will receive attention.

What should I avoid doing?

These heuristics are not to be used as rules to follow blindly as some may or may not apply to your specific scenario. The best approach is to consider each heuristic individually, determine whether it is applicable to your situation, and weigh the results according to how important a problem is to your overall system design.

4.3 For the designer: A set of heuristics for mitigating mode confusion

In this section, we present fourteen heuristics that are designed to mitigate mode confusion. The heuristics have been grouped under four general headings, each representing a principle of human-computer interaction: visibility, feedback, constraints, and consistency. Note that the order in which these heuristics and groupings are presented is not meant to suggest any weighting; each heuristic may be equally important in the development of a user interface.

A Visibility

Concisely, visibility is the amount of exposure something is given. If something is highly visible, then it has been giving a lot of emphasis. If it has low visibility, then it is buried away or completely hidden. Providing the right level of visibility can be challenging as having too many high-priority items can cause desensitization while something with too little priority could be missed by the user.

A1 Give important features more visibility

Give mode annunciators, primary controls, and important system messages higher visibility than less important features. This ensures that the user focuses on the critical parts of the system and does not get distracted by items of secondary importance. Common ways to give a feature more visibility are to: change its size, shape, or colour; use a modal dialogue box; or add auditory feedback. Be careful if you are using a different colour (like red) to show that that an option is dangerous because if everything else on the screen is drab and only your cautionary text stands out, the user's eyes will actually be drawn to it because it has more visibility.

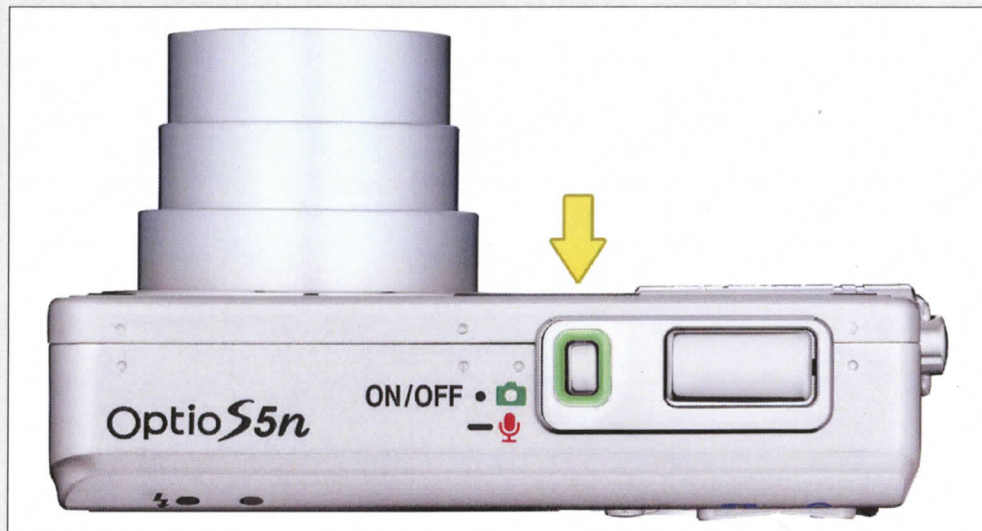


Figure A1: A case of form over function? The Pentax Optio S5n features an illuminated ring that glows green when the camera is in the capture mode, and red during voice memo recording. The problem is that the ring surrounds the power button, not the mode switch. Although this brings attention to the power switch – in a very fashionable manner – neither the capture nor voice memo recording functions are related to power control.

A2 Associate frequently used functions with frequently used buttons

Frequently used buttons are typically placed in optimal locations in the user interface and given special treatment to enhance their visibility. Associating commonly used functions with frequently used buttons helps to channel your user's attention, providing your most salient system features with a greater degree of exposure. Your system should be able to withstand a certain degree of

inadvertent activation of these features, however, as their prominent location means a greater likelihood that these features may be triggered unintentionally.

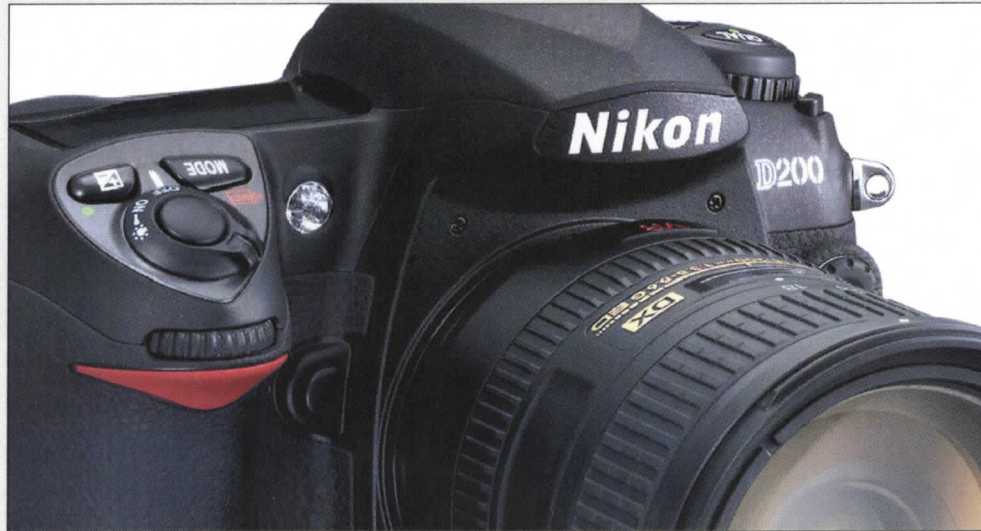


Figure A2: The shutter release button of the Nikon D200 digital camera (round button on the angled face above triangular mark) is conveniently placed underneath the right index finger when the camera is held using the rubber-textured handgrip.

A3 Associate infrequently used functions with infrequently used buttons

Contrapositively, uncommonly used functions – or functions that have the potential for serious consequences if activated inappropriately – should be relegated to less accessible controls. As infrequently used buttons are typically placed in sub-optimal locations that require additional mental or physical effort to activate, it is less likely that a user will activate these functions unintentionally. If safeguarding against improper activation is particularly critical, consider using this heuristic in combination with a flow block, as described in heuristic B2. See §3.3.3.



Figure A3: The button with the light bulb icon is used to activate the Nikon D70 digital camera's top-deck LCD panel illumination. It is positioned slightly out of reach of the user's right index finger when the camera is held by its handgrip, but this is not an issue as the illuminator is designed for use when the user has the camera at waist-level. In such a position, the button can be easily operated using the right thumb.

A4 Use unique adornments to differentiate between interface modes

Differentiate user interface modes from one another by presenting each screen in that mode with some unique adornment, such as a change in coloration, window decoration, or title bar text. The chosen attribute need not be overwhelming, but should not be so subtle that it is inconspicuous. By making each interface mode unique, a user will be able to tell quickly whether they are operating in the intended mode and, if a mode change was requested, be able to confirm this shift in operational mode.



Figure A4: The HP PhotoSmart R707 uses a colour-coded title bar and icon to announce the current operating mode to the user. No colour or icon is used more than once.

B Feedback

According to several researchers [34, 40], the most common cause of mode confusion is the loss of situation awareness. In ideal circumstances, the user would always know what the system was doing, be able to anticipate what the system was going to do next, consistently make the correct choices when presented with several options, and be able to take control if the automation behaved improperly. Although it may not be possible to design your system with such robustness, consider the following heuristics to help a user maintain situational awareness through appropriate and timely feedback.

B1 Let the user know what is happening

Keep your user informed about what is happening by providing appropriate feedback in a timely manner. By keeping them “in the loop”, you give users a better chance of being able to follow the automation flow. They will also have a better understanding of what is expected of them when their input is required or if they need to step in when something goes wrong. The exact way in which this feedback is conveyed will vary depending on the design of your system but make certain that the method you choose is concise, unambiguous, and truthful.

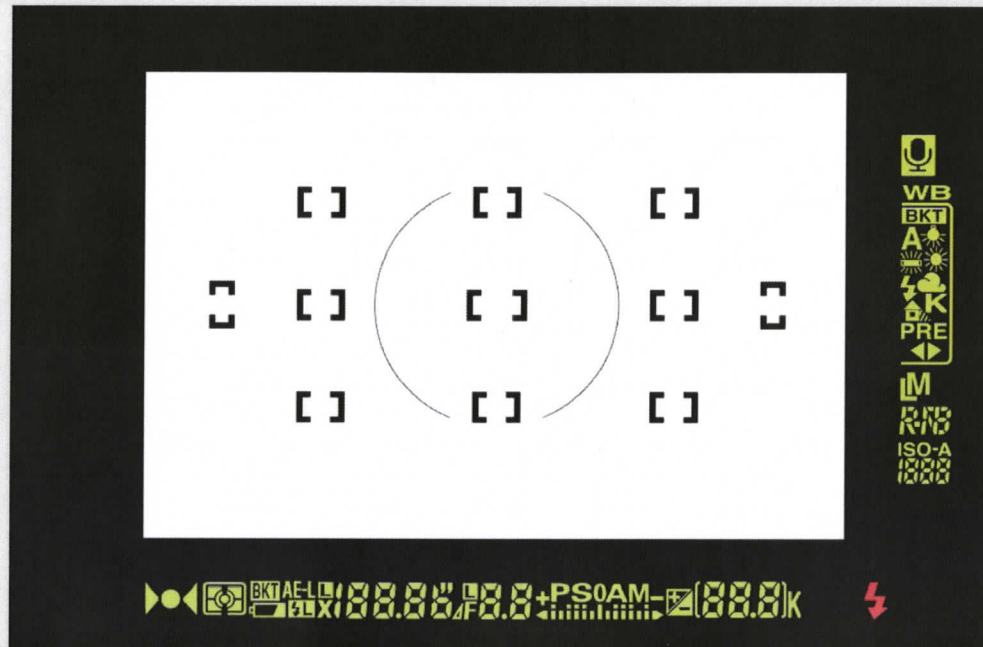


Figure B1: The viewfinder display of the Nikon D2Hs digital camera provides the photographer with a wealth of information. Nearly every feature of the camera is announced by icons in the viewfinder, reducing the need for the photographer to consult any of the three other status displays on the top and back of the unit.

B2 Let the user know what is going to happen next

Providing the user with clues or hints of what is to happen next will give them the ability to anticipate and predict future problems, with the advantage of being able to back out of a situation – if things are not going the way they expected – before it is too late.



Figure B2: Many online e-commerce systems provide a progress bar to let the users know where they are in the checkout process. This clickable checklist allows users to monitor their progress and jump back to a previous point in the process if there is need for revision.

B3 Use indicators to show non-default behaviour

If a feature has been activated that will cause the automation to deviate from its normal behaviour, let the user know so that he or she will not think that the automation is malfunctioning. Even if the change in behaviour is at the user's request, display an indicator as acknowledgement and as a reminder for the user. In cases where there is no default behaviour, use an indicator at all times to let the user know which behaviour to expect from the system.

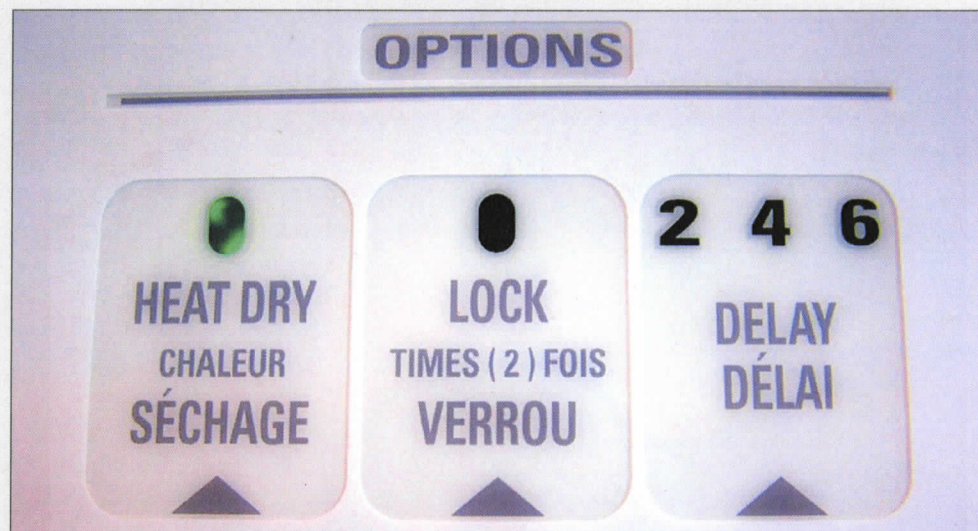


Figure B3: This dishwasher has a delay feature that lets you postpone the wash cycle for two, four, or six hours. The number corresponding to the delay lights-up to let the user know that the dishwasher is operating properly and that this delayed behaviour is intentional.

B4 Use alerts and dialogues judiciously

Studies have shown that humans are only capable of handling a limited number of tasks at the same time [46]. If your user is already juggling several things at once and you present a dialogue requesting the user's attention, you are likely to exceed the cognitive capacity of the user, leading to mistakes. To prevent this mental overload, consider only using high-priority alerts – such as modal dialogue boxes – for serious conditions when interruption is absolutely critical. See §3.7.6.

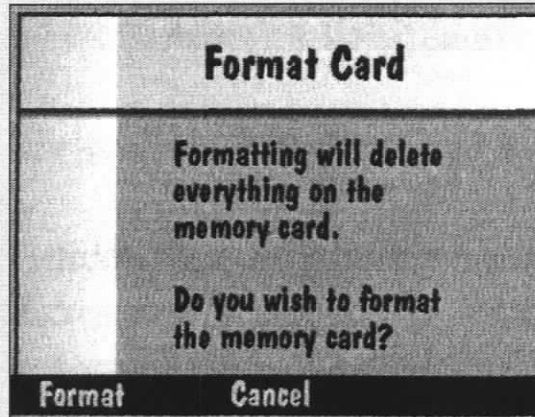


Figure B4: The memory card formatting function of the Kodak DC265 Zoom digital camera uses a modal dialogue box to confirm the format operation before execution.

C Constraints

Several researchers [30, 12] have suggested that usability engineering is as much about controlling the user as it is about designing a good system interface. By judiciously limiting the user's ability to affect change in a system, you can direct the user towards making a correct choice, taking an appropriate measure, or forcing the user to contemplate a potentially dangerous action before commitment. See §3.7.4.

C1 Prohibit a single control from changing more than one variable

Allow a single control to change only one variable at a time. If a control changes multiple parameters simultaneously, the user might be unaware of secondary effects as they may only be focusing on a particular result. Sticking to a one-to-one relationship assures that a control is only activated with a single purpose in mind. There are two acceptable situations, however, where a single function can affect multiple variables simultaneously: when a control is used to change the operating mode of the system; or to reset several variables back to their default values. See §3.7.5.

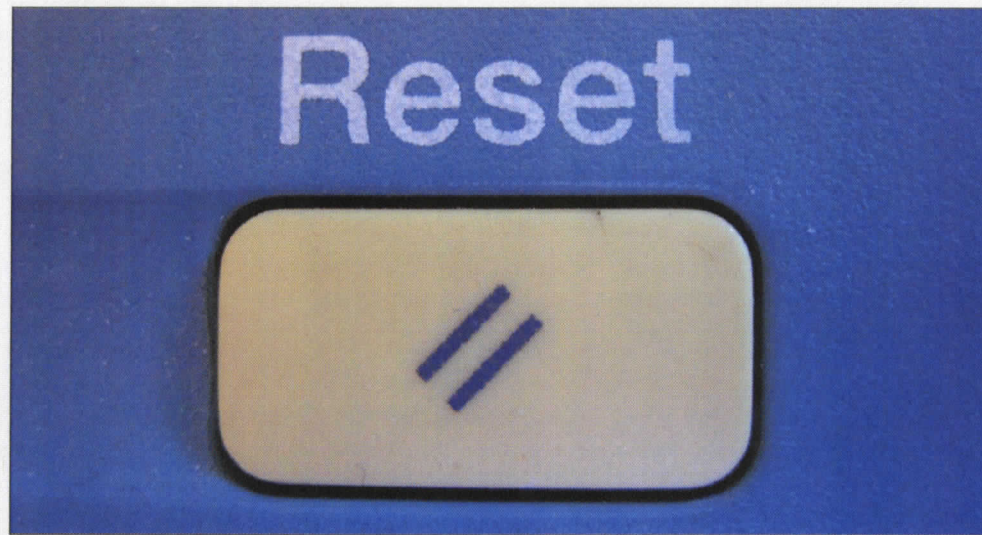


Figure C1: The “Reset” button on this photocopier is used to return all paper handling and reprographic parameters back to their default settings, as specified by the designer.

C2 Make potentially hazardous functions harder to activate

If a feature has the potential to do grave damage if used improperly, make it harder for the user to activate. This will not prevent mode confusion from occurring, but it will force your user to stop and re-think their action before committing to it. You can make things a bit hard by using a simple “OK / Cancel” confirmation dialogue box, or to really make things tough, force the user to enter some sort of random text string or button combination before continuing. See §3.3.3.

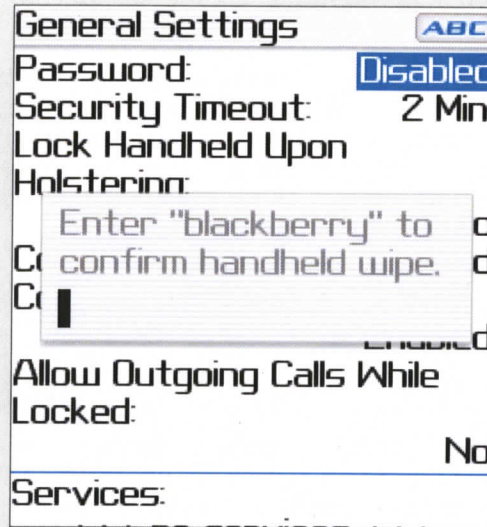


Figure C2: To completely wipe the contents of a BlackBerry 7130c wireless handheld device and reset the unit to its factory defaults, the user is required to key in the word "blackberry". This feature is designed to prevent inadvertent activation and the complexity of the solution underscores the serious nature of the operation.

D Consistency

Providing the user with a user interface which behaves in a similar manner from screen to screen will help a user leverage the lessons learnt in one part of the user interface in other areas. Do not interpret consistency as duplication, however, as users may become confused if two screens look too similar.

D1 Distinguish between features that can and cannot be performed

If a feature is unavailable, remove all indications and disable affordances in the user interface that may suggest otherwise. In a graphical user interface, indicators and menu options for functions which are unavailable can remain in place but should be greyed out to indicate that an option is not permissible in the current operating context. For mechanical controls, consider using an annunciator light to notify the user that the control is inoperative. If an annunciator light is typically illuminated when an operation is enabled, extinguish the light when the control is disabled.

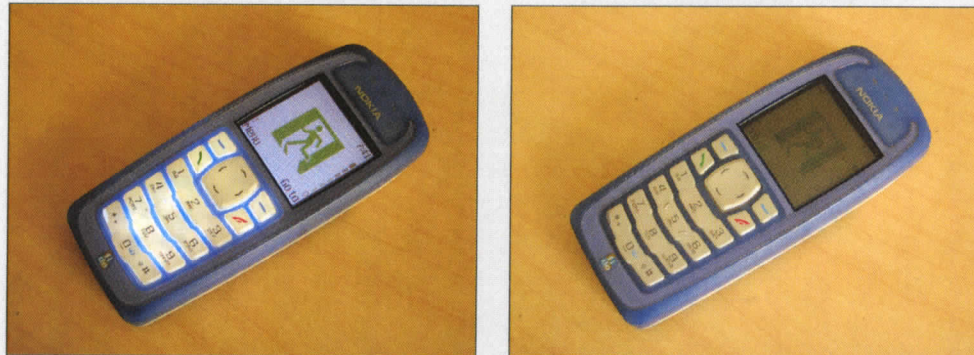


Figure D1: When the Nokia 3100 mobile telephone is ready for use, the keypad is automatically backlit whenever a button is pressed (L). When the keyguard is engaged, however, the mobile rejects all input and the keypad stays dark (R), letting the user know that the keypad is inoperative.

D2 Maintain a consistent spatial layout within modes

To help the user maintain system context, provide a consistent spatial layout (i.e. visual organization) within each mode. Disabling an option, but leaving placeholders or gaps to maintain spatial layout, is favourable to removal as novices who are unfamiliar with the system can note the location of disabled options for future use. If screen “real estate” is at a premium, unavailable items may be completely removed from the interface but at the expense of some situational awareness and didactic ability. See §3.3.1.

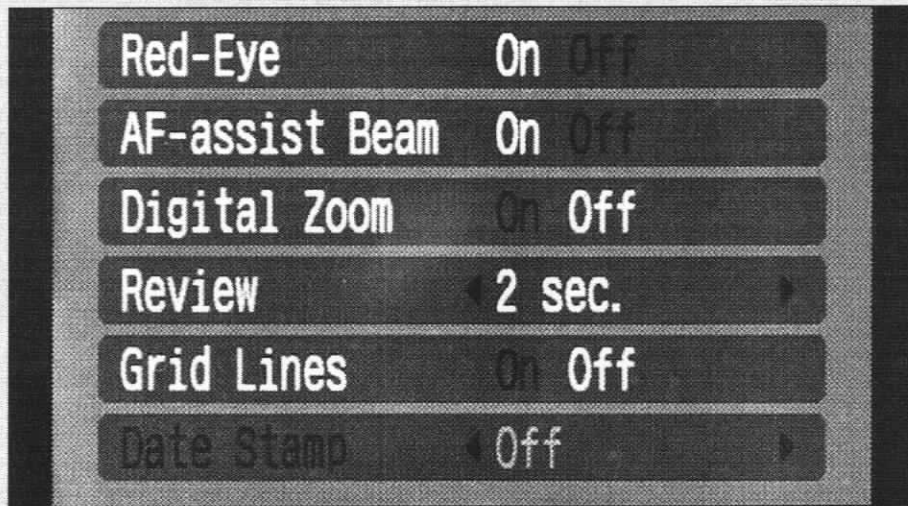


Figure D2: An example of a disabled function in the Canon PowerShot A700. The text for the "Date Stamp" function has been greyed out and the menu selection bar skips over the selection when navigating through the menu.

D3 Use only one menu design paradigm per interface

Using multiple menu presentation techniques may seem like a natural choice to avoid confusion between modes but this scheme actually increases the cognitive workload of the user as each paradigm requires a different interaction technique. Furthermore, the interaction behaviours used in each could be incompatible between paradigms, resulting in a greater propensity for mode confusion. Limiting the user interface to one interaction scheme ensures that the user will access functions in a consistent and predictable manner.

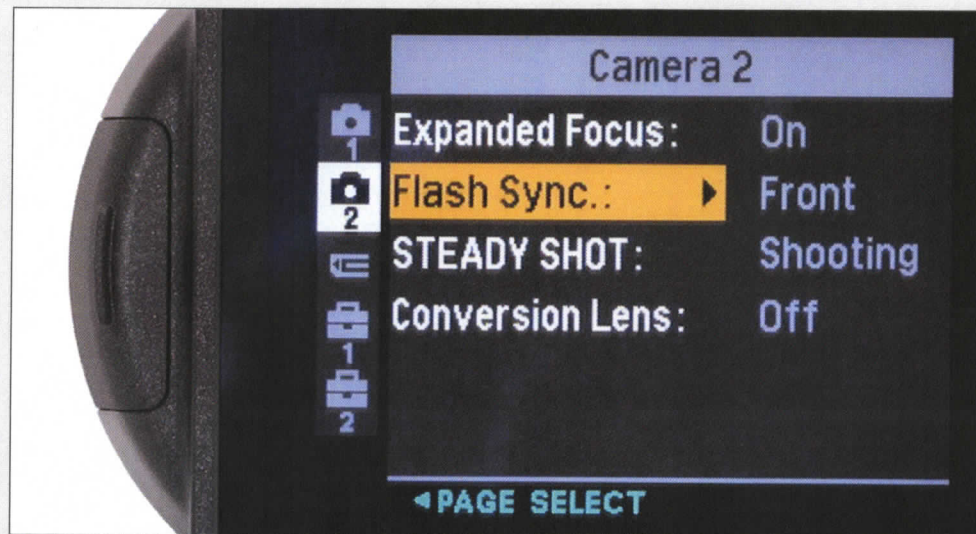


Figure D3: The user interface of Sony CyberShot H5 uses the same paradigm from menu-to-menu. In fact, Sony uses the same menu scheme across its entire line of CyberShot digital cameras and HandyCam video cameras. In theory, this should allow the user of one device to transition to another model or type with less training and reduced cognitive overhead.

D4 Ensure that related operations behave in similar ways

Functions that are semantically similar should be activated in syntactically similar ways. As it is natural for a user to apply the behaviours learned in one portion of the user interface to other areas which are logically similar, design a system that takes advantage of this behaviour by providing the user with a consistent interface that facilitates skill reuse.



Figure D4: The zoom and focusing rings on the Canon EF 24-70mm f/2.8 L USM lens operate in similar manners: for a distant object, turn the zoom ring counter-clockwise to zoom in and turn the focus ring counter-clockwise to focus. In photographer's parlance: turning the zoom ring counter-clockwise increases the lens' focal length; turning the focus ring counter-clockwise moves the focal point towards infinity. In both cases, the "numbers" for both parameters become larger with a counter-clockwise turn.

Chapter summary

In this chapter, a set of heuristics were described for alleviating mode confusion, based on the theories §2, to avoid the errors of §3. In the next chapter, these heuristics are adapted to the domain of digital cameras and applied to retail consumer products in order to evaluate the heuristics' ability to identify situations with mode confusion potential. In §6, these scenarios are then compared against actual instances of mode confusion identified during a user study performed to determine the heuristics' ability to forecast designs with mode confusion potential.

5

Application of the heuristics to digital cameras: heuristic evaluation

In the previous chapter, we described a set of heuristics designed to help a system engineer with a demonstrated expertise or little experience in the study of human-computer interaction to identify scenarios with mode confusion potential. In this chapter we will briefly revisit our research goal (§5.1) and research question (§5.2), both introduced in §1, and describe the two evaluative processes (§5.4 and §5.5) we used to determine the effectiveness of these heuristics when applied by users who may not have formal human-computer interaction training.

5.1 Research goal

Succinctly, the goal of this research is to improve the usability of digital camera designs. The rekindling of interest in personal photography during the past decade has been largely attributed to the development of low-powered, low-cost imaging sensors that have enabled the mass production of digital cameras at a cost favourable to consumers. As digital cameras represent a large departure in technology from their film predecessors, many best practices for digital camera design have yet to be established. The development of film cameras, in comparison, may have reached a plateau more than ten years ago with products that have yet to be bested. As the digital photography market is very lucrative – with a forecasted revenue of 6.8-billion US dollars from the sale of 29.5-million units during 2006 in the United States

alone – manufacturers of digital cameras have been entangled in a long-running “feature war” to gain market share through the development of new technologies. While sensor resolutions, focal lengths, and apertures are quantifiable attributes that can be easily compared and marketed to consumers [16], aesthetics and product usability are nevertheless important factors that digital camera manufacturers must address to create successful products.

Usability testing is a common technique used by designers to determine how well sample users, drawn from the target audience, can use the system to fulfill its intended purpose. This technique identifies usability issues in a system and designers can use this information to refine the product, in iterations, to improve usability and operator efficiency. While this type of testing is very effective at uncovering usability issues, it has several logistical disadvantages. First, usability testing typically requires a working prototype of the system for end-user evaluation. Prototypes are costly to build and represent a stage of design that is quite late in the development process. By the time a prototype for a system can be assembled, many important design decisions have already been made and if a significant usability issue is identified in the core of the design, it could be very costly or even impossible to remedy. Second, usability testing is a time consuming process that requires the coordination of designers and users. Before a usability test can be conducted, a testing regime must be developed that will guide the users in performing tasks that the engineers believe will draw out usability problems in the design. These tests must be conceived with due deliberation as the quality of the results are directly related to the questions posed by the researcher. Finally, usability tests can be prohibitively expensive as the aggregated cost of developing a prototype, recruiting users, and removing staff from active development can be considerable, making the business case for a usability test difficult to justify in a small organization.

The objective of this research is to improve the usability of digital cameras by developing a faster and cheaper evaluation technique that can be used earlier in the design process to

identify and mitigate scenarios with mode confusion potential. Identifying and resolving issues in the initial stages of a project can translate into large savings as modifying systems or retrofitting solutions to an established design can be very costly. This effort is not intended to supplant usability testing however, but is rather envisioned as a complementary procedure that could be used early in the design process to resolve essential problems before their effects propagate throughout the system. Usability testing would continue to be the final measure of product usability, though by resolving a number of issues upfront, it is hoped that the usability testing process can either be shortened, as there would be potentially fewer issues to detect, or made more effective, as designers would now be able to focus on more in-depth problems.

As a first step to this goal, a set of heuristics has been built upon the established principles of human-computer interaction research. To ease the uptake of these heuristics by engineers, one of the primary requirements was to make these heuristics readily understandable and applicable by individuals with little or no formal training in the field of human-computer interaction. The success of this research is contingent upon whether a non-expert user can apply these heuristics easily and if the scenarios with mode confusion identified correlate with actual instances of mode confusion. Heuristics were chosen as the delivery method for these usability guidelines as they can be employed with little penalty during the earliest phases of product development to help guide engineers towards a usable design. Had these heuristics been expressed as a usability certification process or as a rigid developmental framework, a greater emphasis may have been placed on the development of a prototype before testing, possibly shifting the balance towards a design that favours form-over-function, and leading to usable but possibly ineffective products. The researcher believes that heuristics offer a good balance between the technical and ergonomic requirements for a digital camera design, bringing both usability and technical concerns to the foreground so that they may be evaluated simultaneously as peers during the system design process.

5.2 Research question

While the heuristics described in the previous chapter are derived from well-grounded and accepted principles of human-computer interaction [9], the heuristics have been modified and adapted for a specific type of usability error: mode confusion. As these heuristics represent both a departure and an expansion from the original source material to focus specifically on the issue of mode confusion, it was felt that evaluations were needed to confirm the power of these heuristics to identify scenarios with mode confusion potential. Specifically, our research question was to determine what kinds of errors a designer with little or no human-computer interaction experience would uncover using these heuristics.

Two evaluative techniques were used. In the first, we asked a specially recruited evaluator with no formal training in human-computer interaction to apply the heuristics and identify scenarios with mode confusion potential within a set of three sample digital cameras. As a designer of a digital camera would most likely use these heuristics as part of a formative evaluation, we simulated this process by restricting our evaluator's exposure to the instruction manuals of the sample cameras. The second technique, a summative evaluation, was performed as a seven-participant usability study where our goal was not so much to uncover mode confusion errors but to see if the heuristics had previously identified these errors and, moreover, to determine if these heuristics had uncovered potential errors not found in the usability study. For a discussion of formative and summative evaluation techniques, refer to §2.5.1 and §2.5.2.

5.3 Evaluation process

The evaluative process was comprised of four distinct phases. In the first phase, a collection of five digital photography use cases (§5.3.1) and a sample set of consumer-oriented digital camera models (§5.3.2) were collated to establish a common scope and platform upon which to execute the evaluation. During the second phase, an expert user was recruited as an

evaluator to apply the heuristics described in §4 to each model of camera for identifying scenarios with mode confusion potential (§5.4). During this process, the evaluator was not given access to the digital cameras themselves and was only provided with the operating instructions as furnished by each camera's manufacturer. As these heuristics are designed for use early in the development process – perhaps even before a prototype has been built – we wanted to simulate this limitation by providing the evaluator with only a mental concept of each camera model rather than a fully developed and operational product. During the heuristic evaluation, the evaluator uncovered several scenarios with possible mode confusion potential that were not covered by the original set of use cases and the decision was made to extend the list of use cases to provide coverage for some of these esoteric functions. The researcher acknowledges that amendment of the use cases introduces a bias to the usability study but after due consideration and inspection felt that the additions were both reasonable and prudent for the sample set of digital cameras as they cover obscure, but crucial parts of the camera designs. The third phase of the evaluation was a usability study wherein seven participants were asked to perform tasks, as described by the general and extended use cases, to uncover actual situations of mode confusion (§6).

5.3.1 Collating a set of general use cases for digital cameras

Use cases are scenarios of usage that are designed to encapsulate potential ways of utilizing a system's affordances and resources to produce some desired effect. In formulating a set of representative use cases, general patterns of usage for digital cameras were identified and a group of core scenarios were synthesized that the researcher believes to describe accurately the common uses of digital cameras. It is understood that the scenarios identified do not cover the entire feature set of each individual camera and that this aggregation represents only the most basic functions generally available in consumer-oriented digital camera designs.

UC1 Turning the camera on and off

Though power control is a rudimentary and essential function, there exists no standard for powering up or powering down digital cameras. Even amongst offerings from the same camera manufacturer, a variety of designs are used to control the powered state of the system: time-delayed momentary push buttons (spring-loaded buttons with must be held depressed for a certain length of time before activation); momentary slide switches (similar in operation to time-delayed momentary push buttons except oriented in a horizontal plane); or indexed switches (switches that feature a selector that can only be moved to a fixed number of positions). In addition, many digital cameras also feature an automatic off or “sleep” function that puts the camera into a low-power standby mode after a certain period of time elapses with no user interaction. The goal of this task was to determine whether these power switch designs or standby behaviours could lead to mode confusion.

UC2 Taking a picture

The purpose of the second task was to evaluate the general picture taking process for instances of mode confusion or mode confusion potential. Here, a need for a controlled environment was identified to ensure testing integrity. As a camera’s exposure metering system determines aperture and shutter speed settings by measuring the quality of light in a scene, a camera will automatically make compensations for darkened environments by either adding supplementary light (if equipped with a flash unit) or increasing exposure durations. As there is some variance in low-light behaviour between camera models, a well-lit subject in a bright environment was chosen to normalize the test environment so that these compensations would not be necessary. Before this task was performed, each camera was powered on and put into the still photography mode and any user customizable settings were returned to factory defaults. The shooting mode selector was put into the fully automatic mode (or an equivalent mode, with the greatest amount of automation, if a fully automatic mode was not available).

UC3 Using scene modes

Many digital cameras include *scene modes* that optimize the camera's automatic operation to suit particular types of subjects, e.g. portraits, landscapes, close-ups, night scenes, or fast-action sports. In addition to adjusting the camera's exposure determination algorithms, scene modes can also enable, disable, or otherwise modify the behaviour of the camera's subsystems by setting focusing modes, flash modes and transport modes, as pre-determined by the system designer, to produce the desired effect. The advantage of scene modes is that it allows users with little technical knowledge about photography to capture an image in a particular style by simply informing the camera of the desired effect and letting the camera decide on the best settings to use. The object of this task was to determine whether there are any disadvantages to using scene modes, specifically focusing on how scene modes inform users of subsystem mode changes.

UC4 Reviewing photographs

Perhaps the most compelling feature of digital photography is the ability to review images immediately after capture. With early camera models, users were required to move a sliding selector to switch between the image capture and picture review modes, effectively partitioning capture-related functions (such as adjusting camera exposure) and review-related functions (such as deleting images) from each other. In many newer designs, the sliding switch has been retained but a "quick review" feature has been added which allows for the recall of the last recorded image without the need to change the sliding switch position. The goal of this task was to determine whether the quick-review feature contributes to mode confusion.

UC5 Deleting a single photograph and deleting all photographs

Another strong reason for the succession of digital cameras over their film counterparts is the ability to delete undesired images immediately, or later upon review, to recover storage space allowing for the capture of additional photographs. The only appreciable disadvantage to this ability is that images can be deleted (either singly or in bulk) by mistake. Although this may be an infrequent occurrence, there is some anecdotal evidence that it has happened to most users at least once and sometimes with potentially disastrous consequences if the images are of importance. As the inadvertent deletion of a single image or all images simultaneously is a significant event, the researcher felt it was important to include this task in the heuristic evaluation to determine if mode confusion could be a factor in deleting images unintentionally.

5.3.2 Selecting digital cameras used in the evaluation

Three consumer-oriented digital camera models were used during this evaluation — the Canon PowerShot A40 [4], the Canon PowerShot S30 [5], and the Olympus Camedia C-3040 Zoom [10]. Additional models were available but only three were chosen because it was felt that using additional models would not yield appreciably different results, as the purpose of the study was to identify the effectiveness of the heuristics rather than to evaluate cameras against each other. In the event that the results were inconclusive and it was believed that a more expansive selection of cameras would have yielded more meaningful results, further efforts may have been undertaken. At the conclusion of the usability study, however, the quality of results indicated that the examination of further camera models was not required to measure qualitatively the accuracy of the heuristics.

5.3.3 Description of digital cameras

The digital photography market is very competitive. The ability for a company to differentiate a product from models offered by the competition, as well as amongst its own

offerings at different market strata, is extremely important and designs must target specific market segments to be successful. The results of this diversification are designs that feature one or more unique elements of novelty or utility that are particular to a camera design. In this section, each camera used in the evaluation will be briefly described in turn, with emphasis placed on any aesthetic or functional qualities that set this camera apart from other contemporary designs.

Canon PowerShot A40

The most basic member of Canon's A-series of entry-level digital cameras, the Canon PowerShot A40 (Figure 5.3.3a) is designed for inexperienced users with only a passing familiarity with film and digital cameras. As such, ease-of-use and point-and-shoot operation are critical considerations and the PowerShot A40 represents Canon's best effort at producing an inexpensive model that combines intuitive operation with a basic set of features. Because worry-free operation is a driving factor for the camera's target audience, the PowerShot A40 features only three operational modes — two automatic exposure modes and one with manual control. Any semblance of manual control is rarely seen in a camera at this price point and the exceptionality of its inclusion is underscored by the fact that this mode has been eliminated from the PowerShot A40's immediate successor and all subsequent iterations of Canon's low-end A-series product line.

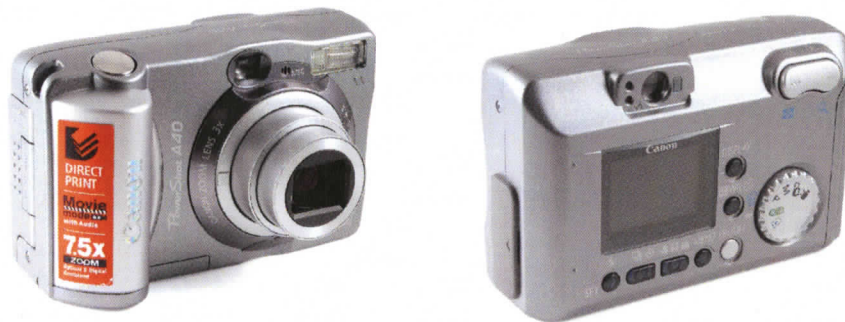


Figure 5.3.3a: The Canon PowerShot A40 digital camera is an entry-level digital camera that combines a basic feature set with easy operation.

Canon PowerShot S30

The Canon PowerShot S30 (Figure 5.3.3b) is an intermediate-grade digital camera designed for fashion-conscious users who are willing to make some ergonomic concessions for the sake of aesthetics. The S-series of cameras are designed to be simple, sleek, and stylish: the simplicity aspect is best underscored by the generously sized mode dial atop the camera that features not only a fully-automatic mode for point-and-shoot photography, but also several scene modes, known as “Image Zone” modes in Canon parlance, which fine-tune the camera’s automatic operation to suit shooting particular types of subjects: portraits, landscapes, close-ups, night scenes, and fast-action sports. Though its design favours form over function, it nonetheless includes a “Creative Zone” that provides the option of manual control for certain features, such as focus, shutter speed, and aperture, giving the operator more control in creative photography.



Figure 5.3.3b: The Canon PowerShot S30 digital camera features a sliding lens cover that performs three functions: to turn the camera on and off; to switch between capture and playback modes; and to protect the lens from damage when the camera is not in use.

The most striking design element of the S30 is the sliding lens cover that performs a trio of functions: to act as a lens protector; to power up the camera from a “soft off” state; and to change the camera mode from stills recording to playback.

Olympus Camedia C-3040 Zoom

As Olympus' top-of-the-line consumer digital camera model, the Camedia C-3040 Zoom (Figure 5.3.3c) is the most feature-rich model of the sample selection. This camera offers users a selection of automatic and manual exposure modes as well as scene modes to assist novice users with capturing images in differing environments. The camera is solidly built and includes a generous assortment of buttons and controls but some are awkwardly placed and spread out along the length of the lens barrel. To help the user locate features, hepatic hints – such as raised portions or a different button surround – are included for each button, giving users familiar with the camera the ability to distinguish buttons from one another by touch; less familiar users will need to constantly refer to the button labels to find particular functions.



Figure 5.3.3c: The Olympus Camedia C-3040 Zoom digital camera provides the advanced photographer with a variety of automatic and manual controls, making it ideal for creative photography.

5.4 Heuristic evaluation

The purpose of the heuristic evaluation was to use the set of heuristics described in §4 to identify situations of potential mode confusion in the sample set of digital cameras within the scope of the use cases outlined in §5.3.1. In particular, we were interested to see if the evaluator, who has no formal training in human-computer interaction, would be able to apply effectively the heuristics to the camera designs.

5.4.1 Recruitment of an expert user as “evaluator”

In ideal circumstances, a product engineer with an established history of digital cameras design would have been recruited to apply the heuristics to the sample set of camera models. The researcher contacted several industry-leading camera manufacturers in hopes of establishing an open dialogue and information exchange, but unfortunately, no such opportunity presented itself. As an industry resource was not available for consultation, an individual familiar to the researcher who was deemed to have an expert knowledge in the general operation of consumer-oriented electronic devices – and was cognizant of, but did not have any formal training in human-computer interaction or human-factor engineering – was explicitly recruited to apply the heuristics described in §4 to each digital camera model.

Evaluator

Gender: Male

Age: 30-40

Years of experience with digital cameras: More than five years

Familiarity with digital cameras: Expert

The evaluator is a senior programmer analyst at the University of Victoria. Responsible for the procurement and training of faculty and students on many types of multi-media equipment, this participant is considered an expert in the use of digital cameras but does not have any formal training in human-computer interaction or system design.

5.4.2 Formative evaluation of the heuristics

To determine whether the heuristics could be readily applied by a non-expert in human-computer interaction, we simulated a formative evaluation (§2.5.1) by providing the evaluator with a copy of the heuristics (in a form similar to those presented in §4.3), the use cases described in §5.3.1, and electronic copies (Adobe PDF files) of the instruction manuals for the three sample digital cameras. The evaluator was not given physical access to the

digital cameras themselves. The evaluator was then asked to become familiar, to the best of his ability, with the operation of each digital camera using the information provided in the instruction manuals and then to consider each heuristic using, but not limited to, the general use cases collated by the researcher.

The findings of the evaluator have been paraphrased in §5.4.3. During this evaluation process, the researcher was available for consultation in person, by telephone, or via electronic mail, but limited himself to answering questions pertaining only to the evaluative process and heuristics, refusing to comment on specific digital cameras, the implementation of features, or other topics that may have biased the evaluator's findings.

5.4.3 Heuristic evaluation results

The following results have been distilled from the evaluator's observations. The heuristic used to identify each issue are as noted by the evaluator while the comments are the contribution of the researcher.

Canon PowerShot A40

Heuristic: B1 – *Let the user know what is happening*

B2 – *Let the user know what is going to happen next*

B3 – *Use indicators to show non-default behaviour*

Issue: “While the flash is charging, the shutter is locked. Users may interpret this off-line period as a malfunction. Perhaps a message should be displayed to the user to let them know why they cannot take a picture.”

Researcher's comments: In this scenario, the evaluator has attempted to take a photograph with flash before the flash capacitor has had a chance to charge. As a result, the camera has “locked” – i.e. disabled – the shutter release button until the flash subsystem has stored up

enough energy to produce a proper exposure. System engineers often use interlocks in designs to ensure that sequenced operations are done in the correct order, but typically inform the user to let them know of this requirement; in this case, however, the user has not been advised of the interlock behaviour creating a scenario with mode confusion potential. See §3.7.4 for a discussion of interlocks and lockouts.

*

**

Heuristic: C1 – *Prohibit a single control from changing more than one variable*

Issue: “Changing mode dial between M[anual] and A[utomatic] changes manually selected flash settings.”

Researcher’s comments: This is an example of an indirect mode change. The evaluator’s intention was to change the camera’s mode from manual to automatic operation while retaining a manually selected flash setting (e.g. flash disabled). Unfortunately, this is not possible, given the system design, as a switch to automatic operation will reset the status of all manually selected features. Unless explicitly informed, users should be given the opportunity to alter variables individually; if a user can change multiple variables using a single control, the user may not realize the scope of their changes, leading to possible mode confusion.

*

**

Heuristic: A2 – *Associate frequently used functions with frequently used buttons*

A3 – *Associate infrequently used functions with infrequently used buttons*

Issue: “Bi-directional rocker switch is confusing because user must press the ‘Set’ button to change from left-right movement to up-down movement. Directional buttons should have been included for all four directions.”

Researcher's comments: An interesting usability issue of this camera model is that the directional control pad (Figure 5.3.3a), located immediately below the liquid crystal display, features only two buttons – left and right – but the user interface is built with four directions in mind. Access to the “up” and “down” features require the prior activation of a “shift” button to change the orientation of the bidirectional pad from horizontal to vertical. As use of the “up” and “down” directions are implied as strongly as the “left” and “right” directions in the user interface, users may be confused by the lack of visibility for these two directions.

*
**

Heuristic: A4 – *Use unique adornments to differentiate between interface modes*

Issue: “The image protection feature is confusing because the ‘key’ icon always appears in the top-left corner, suggesting that the image is protected. In actuality, the user needs to press the ‘Set’ key to enable the protection feature.”

Researcher's comments: This feature is problematic because in addition to using the “key” icon as the adornment for the title bar of the image protection mode, the same icon is used to indicate that the image protection feature has been enabled. This ambiguity can lead to an interface interpretation error.

*
**

Heuristic: D1 – *Distinguish between features that can and cannot be performed*

Issue: “Users may believe the camera’s power is turned on because they can see through the viewfinder even though it is actually powered off. What if some type of viewfinder blind was added to the design?”

Researcher's comments: This camera design features an optical viewfinder that remains usable at all times, regardless of the powered condition of the system’s electronics or operational mode. Such an arrangement can lead to scenarios with mode confusion potential

if the user employs the optical viewfinder as the sole determinate for the powered state of the camera. By being able to see through the viewfinder, the user may incorrectly believe that the camera is turned on even though it is actually powered down.

Canon PowerShot S30

As the PowerShot S30 uses a user interface similar to that of the PowerShot A40, several issues with mode confusion potential, such as the locked shutter button while the flash unit is charging, ambiguity in the image protection feature, and indirect mode change between manual and automatic modes are similarly applicable to the PowerShot S30. For brevity, these issues have not been reported again; refer to the previous subsection for a discussion of these topics.

Heuristic: B1 – *Let the user know what is happening*

C1 – *Prohibit a single control from changing more than one variable*

Issue: “The interaction between the sliding lens cover and the sliding playback switch on the back of the unit is confusing. Both the lens cover and playback switch are used to turn the camera on and off or change from capture to playback mode.”

Researcher’s comments: The interaction between the sliding lens cover and sliding playback switch in this camera design was found to be so complex and confusing that a section was added to the post-study questionnaire portion of the usability study to further explore this topic; study participants were asked to draw a diagram showing the relationship between these two controls and an annunciator, a dual-colour capture ↔ review mode indicator light. See Appendix D for a detailed discussion of this unique mode confusion problem.

Olympus Camedia C-3040 Zoom

Heuristic: B1 – *Let the user know what is happening*

B3 – *Use indicators to show non-default behaviour*

Issue: “Manual says that the camera automatically enters a standby mode after a period of rest but there is no indication that this had happened. You cannot tell whether the camera is asleep or has been turned off.”

Researcher’s comments: This camera has a standby feature wherein the camera switches into a low-power mode after three minutes of neglect. Any subsequent shutter button press or zoom lever activation wakes the camera from its sleep but it does not register the first action; a user, for example, would need to depress the shutter button twice to take a picture with a dozing camera — the first to wake the camera and a second to take the picture. The cause of this mode confusion issue is insufficient visibility of the system’s operational state; if a change in the power mode annunciator (e.g. change of colour or use of a blinking illumination pattern) was used to represent the camera’s sleeping state, a user would know to expect non-default behaviour rather than mistakenly believe that the camera was ready for use.

*

**

Heuristic: B3 – *Use indicators to show non-default behaviour*

D3 – *Use only one menu design paradigm per interface*

D4 – *Ensure that related operations behave in similar ways*

Issue: “Method for changing between automatic and manual focus modes is inconsistent with other mode selection menus. Manual focus mode uses the ‘up’ and ‘down’ arrow keys to adjust distance, instead of the zoom lever which would have been more consistent with the rest of the user interface.”

Researcher's comments: All options in the main shooting menu of this camera model behave in a similar way except for the focus mode selector. In the other menus, a user navigates "down" a horizontal tree using the arrow keys, presses the "OK" button on the desired option, and the menu system is automatically dismissed. In the case of the focus mode selector, however, pressing the "OK" button on the manual focus option puts the camera into a different user interface wherein the zoom lever is no longer operational and a numeric focal distance must be selected by using the "up" and "down" arrow keys. The decision to use a unique operating paradigm solely for this feature is potentially problematic as the user is not given any indication in the user interface to expect non-default behaviour and, even if such warning were to be given, the change may still pose confusing as the rest of the user interface modes behave in a homogenous manner, creating an expectation for similar operation.

5.4.4 Analysis of heuristic evaluation results

While reviewing the findings of the evaluator, the researcher discovered that, except for one exception, none of the issues identified were found within the use cases developed for the heuristic evaluation. Even though all results fell within expectations (in the sense that later examination by the researcher determined that the evaluator's findings had merit) almost every instance of mode confusion potential identified was found in the more esoteric regions of the sample camera designs. As the use cases were designed to reflect the most common usage scenarios of consumer-oriented digital cameras, this finding suggests that the areas with greatest mode confusion potential are not in the commonly used components but rather in the infrequently accessed features. This evidence also suggests that the probability of a user experiencing mode confusion is low so long as they use the camera within the boundaries established by the researcher's general use cases.

Unfortunately, while users may be content to restrict their use to the common features of a digital camera, there may be circumstances that necessitate the use of these less commonly used – but nevertheless essential – features, thereby increasing the likelihood that they will encounter a scenario with mode confusion potential. As the general use cases were inadequate in uncovering a large portion of the issues identified by the heuristic evaluation, the researcher felt that it was prudent to amend the original set of use cases with a set of directed uses cases which, at a minimum, ensured interaction with the portions of each camera that the evaluator believed to contain mode confusion potential. The researcher acknowledges that amendment of the use cases introduces a bias, but felt it was important to include coverage for these obscure but crucial portions of the camera designs.

One other interesting observation that arose from the evaluation is that these heuristics can be used in both critical and constructive capacities. When applied to a system, whether under construction or after its completion, these heuristics can be used in a critical manner to identify scenarios with mode confusion potential, alerting a designer that changes may need to be made to avoid user errors. Coincidentally, these heuristics also can be used in a constructive role to suggest corrective measures. As an example, consider the first issue reported by the evaluator concerning the Canon PowerShot A40: “While the flash is charging, the shutter is locked. Users may interpret this off-line period as a malfunction. Perhaps a message should be displayed to the user to let them know why they cannot take a picture.” Here, the evaluator has not only identified an issue with potential mode confusion by using heuristic B1 – *Let the user know what is happening*, but has also used heuristics B2 – *Let the user know what is going to happen next* and B3 – *Use indicators to show non-default behaviour* to formulate suggestions that can alleviate mode confusion potential. The researcher believes that using the heuristics in such a manner is particularly powerful as it not only alerts the designer of scenarios with potential mode confusion but also recommends a course of action from which a viable solution can be derived.

Chapter summary

At the beginning of this chapter, we briefly revisited the motivation behind our research and posed a research question, asking whether the heuristics of §4 could be successfully applied by a user without human-computer interaction expertise, which we attempt to answer in two phases. In the first phase, a collection of general use cases (§5.3.1) and a sample set of consumer-oriented digital cameras (§5.3.2) were chosen to establish a common scope and platform upon which to execute the evaluation. In §5.3.3, we provided a description of each digital camera model that discussed, in brief, the target market for each design and some of each camera's most salient features. §5.4.1 through §5.4.3 related the specifics of a simulated formative evaluation, recounted the recruitment of an expert user as "evaluator", described the procedure used to perform the heuristic evaluation, and reported the subsequent collation and analysis of the evaluation data.

§6 describes the second half of the evaluative process — a user study to identify actual situations of mode confusion using a combination of general use cases (§5.3.1) and extended use cases (§6.1).

6

Application of the heuristics to digital cameras: usability study

With the results of the evaluator in hand, the next step in the evaluation process was to correlate these findings with actual instances of mode confusion. As many of the scenarios with mode confusion potential found by the evaluator were esoteric in nature and may not be encountered through the normal operation of each digital camera, the general use cases presented in §5.1.1 were amended with a set of directed uses cases (§6.1) which, at a minimum, ensured that each user would interact with modes of the camera that the evaluator believed to contain mode confusion potential. In general, users were not prescribed a sequence of discrete actions, but were requested to perform tasks that make use of specific features. Some readers may object to this directed style of usability testing, but the researcher believes that though infrequently used functions may not be activated often, these functions are included in a system purposefully and are expected to be used by the user at some juncture. As the reduction of mode confusion potential should be regarded as a general goal of good design practice, the importance of testing an obscure feature for mode confusion potential should be given the same attention as a function with more prominence.

The following subsections discuss the usability study procedure in detail. The extended use cases are listed in §6.1. In §6.2, the recruitment process undertaken to find candidates for the usability study is explained, noting the various advertisement methods used and the participant selection process. §6.3 features a brief dossier of each participant, providing

relevant personal data such as age, gender, experience with digital cameras, and familiarity with digital cameras. All statements gathered regarding the ability of each participant were offered by the participants themselves. §6.4 chronicles the procedure used during the usability study sessions to observe each user and record experimental data. Finally, §6.5 presents the results of the user evaluation.

6.5 A set of extended use cases for exposing specific confusion scenarios

In addition to the general use cases described in §5.3.1, the researcher determined, from the results of the evaluator, that there was a need to provide an additional set of extended use cases to ensure that the study participants would activate the features identified during the heuristic evaluation. As much as could be avoided, users were not asked to follow specific steps but rather were asked to perform tasks that involved specific features. The actions taken by the user to locate, activate and apply each function were unscripted and performed at the volition of the study participant.

Five extended use cases were derived directly from the observations of the evaluator, who aptly identified additional sources of mode confusion potential outside the scope established by the general use cases during review of the instruction manuals. While these extended use cases are expressed in relation to the three sample digital cameras, it should be noted that the usability issues they reference are not limited to these models only and may be present in other digital camera models from the same or differing manufacturers. These extended use cases are not, however, general use cases in the strictest sense but represent a more focused use case directed towards a specific usability issue.

EUC1 Determining whether the Canon PowerShot A40 is powered on or off

In this extended use case, the PowerShot A40 was put into three states: powered off; powered on without LCD monitor engaged; and powered on with LCD monitor engaged. With each

different variation, study participants were asked to determine whether the PowerShot A40 was powered on or powered off, with particular emphasis on using the viewfinder to determine the camera's powered state.

EUC2 Changing focus modes on the Olympus Camedia C-3040 Zoom

Users were asked to familiarize themselves with some of the advanced settings of this camera and asked to use the manual focusing mode, which was previously identified in the heuristic evaluation to be potentially confusing. The goal of this extended use case was to determine whether the incongruent manual focusing mode posed a challenge for participants.

EUC3 Changing between shooting modes on the Canon PowerShot S30

In this extended use case, the PowerShot S30 was first placed into the "P" shooting mode by the researcher. The user was then asked to set the flash system mode setting to the "always on" setting and change the shooting mode dial to the "Tv" (shutter priority), "Av" (aperture priority), or "M" (manual) position — the reason for this extra level of direction is that control of the flash system is not available in all scene modes but is possible in all primary shooting modes. Participants were then asked to take a photograph and discuss whether the resulting flash behaviour was as they anticipated, in light of the previous "always on" setting. The purpose of this extended use case was to observe if users would be confused by indirect mode confusions resulting from the use of multi-variable mode selectors and as a side effect, to observe whether users re-check settings after changing operating modes.

EUC4 Determining if the Olympus Camedia C-3040 Zoom is in standby mode

The Camedia C-3040 Zoom was allowed to sit stagnant for more than three minutes time after which the user was asked to determine if the camera was in a powered up, powered down, or standby state. If the user believed that the camera was in a powered down or

standby state, the researcher asked for possible ways that the camera could be returned to the fully powered state.

EUC5 Using the image protection feature on the Canon PowerShot S30

As the image protection feature was determined to have a strong disposition towards mode confusion in the heuristic evaluation, the researcher asked users to apply this feature to several images to determine whether the presence of the double “key” indicators was a source of mode confusion.

6.6 Recruitment of participants

Before the onset of subject selection, it was assumed that mode confusion would be an issue for users regardless of personal parameters such as age, race, or gender. When deriving the profile for suitable candidates, there was some concern that expert proficiency with digital camera operation may mitigate mode confusion potential. As evidenced by research in the aeronautics domain, even extremely complex systems can be used effectively and in a relatively error-free manner with enough familiarity (either through training or recurrent usage) with the system [32]. Nonetheless, it was assumed for the purpose of this usability study that users of all levels of competence would be affected by mode confusion potential, even if to varying degrees. Thus, it was hoped that subjects with a wide gamete of skill would be recruited to provide a greater degree of coverage.

This preference for variety, however, was not used as a means of discriminating individuals. Participants were recruited by three means: via posting to an electronic mailing list; by posting of printed advertisements in the workplace; and by word-of-mouth (including personal recruitment). Interested parties were asked to respond to the researcher by either email or telephone. In total, seven applicants responded and were subsequently enlisted in the usability study.

6.7 Participants dossiers

As our user study involved subjects who are not active members of the research team, approval of the user study by an ethics committee was required to ensure that the testing procedure was morally sound, that the anticipated cost of participation – in time or inconvenience – was conveyed truthfully to the user, and that the proper releases were obtained from the subjects prior to participation. The University of Victoria has strict policies regarding the use of human subjects [48]:

All research involving human participants must receive approval from the Human Research Ethics Board (HREB). This approval is required both prior to, and during, the recruitment and data collection phases of the study. The mandate of HREB is to ensure that all human research is conducted in accordance with the highest ethical standards and that the public, the researchers, and the University are all protected from harm. University policies and procedures are designed to comply with [The Government of Canada's] Tri-Council Policy Statement on the Ethical Conduct for Research Involving Humans.

A research proposal was submitted to HREB and their analysis determined that our usability testing procedure met the guidelines established by the University of Victoria. A statement of conformance was issued by the University of Victoria Human Research Ethics Board and a copy of this document is included as Appendix B.

For the protection of privacy, any personal information that could be used to identify a participant has been omitted from this thesis. Information with respect to familiarity with and years of experience using digital cameras was provided by the subjects themselves via post-session questionnaires and were not derived from testing or observation by the researcher. Completion of the questionnaire was optional and participants were given the choice to omit any or all questions. See Appendix C for a sample copy of the questionnaire as completed by the study participants.

Participant A

Gender: Male

Age: 40-50

Years of experience with digital cameras: Less than one year

Familiarity with digital cameras: Novice

Participant A is doctoral graduate student at the University of Victoria, Department of Computer Science. The participant notes that he has used film-based cameras in a recreational manner for over twenty years but has never owned a digital camera.

Participant B

Gender: Female

Age: 20-30

Years of experience with digital cameras: Three to four years

Familiarity with digital cameras: Intermediate

Participant B is a Master's student at the University of Victoria, Department of Computer Science who has extensive training and experience in the field of human-computer interaction and a lengthy association with digital cameras. This user's self-evaluation indicates that she is the most experienced user amongst the seven study participants.

Participant C

Gender: Male

Age: 20-30

Years of experience with digital cameras: One to two years

Familiarity with digital cameras: Intermediate

Participant C is a Master's student at the University of Victoria, Department of Computer Science, specializing in the study of software development. This participant has used digital

cameras for several years but does not have any formal training in the study of human-computer interaction.

Participant D

Gender: Male

Age: 20-30

Years of experience with digital cameras: One to two years

Familiarity with digital cameras: Intermediate

Participant D is a Master's student at the University of Victoria, Department of Computer Science. Although this user's area of study is information visualization, this participant has had some formal training in the study of human-computer interaction and usability study procedures.

Participant E

Gender: Male

Age: 20-30

Years of experience with digital cameras: One to two years

Familiarity with digital cameras: Intermediate

Participant E holds a Bachelor's of Science degree from the University of Victoria, Department of Computer Science, and is a programmer by occupation. Although this participant does not have any formal training in the field of human-computer interaction, he has worked extensively with researchers who are active in this discipline and should be considered as a user who is knowledgeable of human factors and user-centred design techniques.

Participant F

Gender: Male

Age: 20-30

Years of experience with digital cameras: Less than one year

Familiarity with digital cameras: Novice

Participant F is a Master's student at the University of Victoria, Department of Computer Science, specializing in the field of knowledge management. This participant has used digital cameras less than one year and considers himself to be a novice user. This user has some formal training in the study of human-computer interaction.

Participant G

Gender: Male

Age: 20-30

Years of experience with digital cameras: Less than one year

Familiarity with digital cameras: Novice-Intermediate

Participant G is an undergraduate student at the University of Victoria who recently completing his third year of study in the Department of Computer Science. This individual indicates that he has a good background in consumer electronics but has little experience with digital photography products.

6.8 Procedure

To protect the privacy of each participant and to ensure that each respondent's answers were their own, each user was interviewed individually. Users were encouraged to speak aloud throughout the usability study, noting their intentions and identifying any factors used to make decisions. These vocalizations granted the evaluator insight into: the learning process of each user as they became progressively familiar with the look and feel of each camera; the way the user developed their mental model of the system's operation; and, if mode confusion occurred, the reasoning of the user at the time of the conflict and the subsequent troubleshooting process used to formulate a resolution.

The usability study sessions were held in a second floor conference room of the Technology Enterprise Facility I building on the University of Victoria campus over the span of several hours. Each session lasted approximately 30 minutes, with the latter half of each session dedicated to the completion of a post-session questionnaire by the participant. Present during the sessions were the study participant, the researcher, and a research assistant who operated a video camera, used as a supplementary observation tool. The researcher used two supplementary observation channels to capture experimental data during the usability study sessions: the visual recording portion of a digital video camera to capture the interactions between the users and sample cameras, and the audio recording portion to log the spoken commentary of the users during the study process. Keeping an audio and visual record of the usability study sessions was important to the data collection procedure as it allowed the researcher to review each user's contribution for additional evidence of mode confusion that the participant may not have realized at the time or omitted from their written testimony. While users were given the opportunity to forbid the use of one or both of these supplementary observation channels without penalty of exclusion from the study, all seven participants gave consent to have their video and audio contributions recorded.

At the beginning of each observation session, the researcher provided the subject with a brief introduction to the usability study, explaining the motivation for this research and providing an overview of the study procedure. The researcher then introduced each digital camera to the participant, describing the most salient features of each design, after which the user was allowed several minutes of unstructured hands-on time to familiarize themselves with the controls and operation of each model. During this introduction and familiarization process, the researcher fielded questions from the user, provided background information and detailed operating instructions as requested. Once the user felt that they were satisfactorily familiar with each camera, the researcher then asked the participant to perform the tasks suggested by the use cases in §5.3.1. Subjects were not required to complete any of the tasks and were

allowed to skip any or all tasks without the need to cite any reason. Users who chose to terminate the usability study sessions prematurely were also allowed to do so, and any contributions made to that point were either retained as supplementary experimental data or completely discarded at the user's discretion. These withdrawal mechanisms were unused, however, as all users completed the usability study process and all experimental data was retained. Questions posed by the user during this portion of the usability study were not answered if the researcher believed that answering the question would introduce a bias or otherwise skew the experimental data; all requests for clarification of instructions, however, were permitted. Following the completion of the five general use cases, the participants were then directed to perform the tasks suggested by the five extended use cases described in §6.1.

For every general use case and extended use case, the researcher described the goal of each task verbally to the user at a high-level, being careful to convey the requirements clearly, but not to disclose or prescribe the steps necessary to accomplish the task. Following each usability study session, participants were asked to complete a questionnaire to provide a written account of their experiences, to identify any actual instances of mode confusion encountered, and to diagram their understanding of the sliding lens cover design of the Canon PowerShot S30. A comparison and analysis of these self-transcribed mental models can be found in Appendix D. The post-session questionnaire was two-pages in length and consisted of six questions plus a free-form area for written comments; a sample of the questionnaire is included as Appendix C.

6.9 Usability study results

At the conclusion of the usability study sessions, the questionnaires were collected for analysis and the audio and video records reviewed to determine if additional instances of mode confusion had occurred. Here, the "running commentary" provided by the users during the study sessions proved to be invaluable as it was possible for the researcher to identify

user-professed instances of mode confusion that had not been noted on the questionnaires. The results of the seven usability study sessions were then aggregated into a single list and arranged into three groups determined by their congruency with the scenarios with mode confusion potential identified in the heuristic evaluation.

Sources of mode confusion identified during the heuristic evaluation and usability study:

- **Canon PowerShot A40:** User was unable to determine whether the camera was powered on or powered off using the viewfinder alone. **All seven users reported mode confusion.**
- **Canon PowerShot A40:** Bi-directional rocker switch was confusing as the user interface appears to be designed for a system with four directions in mind. **One of seven users reported mode confusion.**
- **Canon PowerShot S30:** Indirect change of flash mode settings was not observed by the user when the shooting mode was changed. **Five of the seven users reported mode confusion.**
- **Canon PowerShot S30:** Protection feature confused user as the yellow “key” icon is used as both an interface adornment as well as a feature annunciator. **Six of the seven users reported mode confusion.**
- **Olympus Camedia C-3040 Zoom:** User was unable to differentiate if a stagnant camera was in a powered off or standby mode. **Four of the seven users reported mode confusion.**
- **Olympus Camedia C-3040 Zoom:** Changing focusing mode from automatic to manual resulted in confusion due to inconsistent system behaviour. **Five of the seven users reported mode confusion.**

Sources of mode confusion identified during the heuristic evaluation only:

- **Canon PowerShot S30:** Locked shutter during flash capacitor charging process may cause confusion in user. **No users reported mode confusion.**

Sources of mode confusion identified during the usability study only:

- **Canon PowerShot A40:** Slow response time of interface contributed to mode confusion as user had moved on to next task before the camera could respond. **One of the seven users reported mode confusion.**

With two exceptions, the instances of actual mode confusion recorded during the usability study coincided with the set of scenarios with mode confusion potential identified by the evaluator during the simulated formative evaluation. This correlation suggests that the heuristics, as applied by a non-expert evaluator, are able to anticipate at least some of results of the user evaluation, lending credibility to the effectiveness of the heuristics to identify situations of mode confusion potential. Although the actual number of instances of mode confusion identified by the simulated formative evaluation and usability study were limited, it should be noted that the three sample cameras are commercially available, off-the-shelf products. It is almost certain that these cameras have already been subjected to a battery of user tests and usability metrics before being released to manufacture and the author believes that the ability to identify these usability issues, though limited in number, is a significant finding. If these heuristics had been applied to designs still under development, it is conceivable that a greater number of design flaws could have been uncovered during the simulated formative evaluation and usability study.

Of the two exceptional cases, one source of mode confusion potential was identified by only the heuristic evaluation while the other was noted only during the usability study. This

variation in coverage was expected as it is widely acknowledged that a heuristic evaluation may uncover details missed by the usability study, and vice versa, as the evaluator in a heuristic evaluation is typically an expert user of the system who views the system from a top-down perspective while the participant of a usability study has a limited understanding of the system as a whole and thus approaches the system from the bottom-up.

This divergence also supports the intention of the researcher to promote this set of heuristics for mitigating mode confusion as a complementary evaluation technique to user studies, allowing for greater test coverage. As heuristics and user studies are typically applied during different phases of the development process – with heuristics employed in the formative period when the system is undergoing active development, and user studies performed when the system is nearing completion to ensure its usability by the target audience – the researcher believes that this set of heuristics may allow for the timely elimination of some scenarios with mode confusion potential, if used at an early stage, before they are entrenched in the system and become too difficult or too costly to remove.

Chapter summary

In §5, we described the first phase of our testing methodology in which an evaluator used the heuristic evaluation technique to simulate a formative evaluation as would be done by a designer of digital cameras. The results of the evaluation suggested that the common use cases for digital cameras may not contain many scenarios with mode confusion potential and that the bulk of user errors are experienced in the less frequently used parts of a camera's design. From the data collected by the evaluator, a set of extended use cases were later developed by the researcher and added to the second phase of the evaluation process. The researcher acknowledged that this amendment introduced a bias into the testing environment, but believed that this extension was necessary in order to provide coverage for these atypical usage scenarios.

The second phase of the evaluation process consisted of a usability study in which seven participants were recruited (§6.2) to perform the tasks suggested by the general use cases (§5.3.1) and extended use cases (§6.1) to uncover actual instances of mode confusion. The chapter concluded with the presentation of the results from the usability study and a comparison between the issues identified by the heuristic evaluation and the usability study suggested a partial correlation, implying that the heuristics, as applied by the non-expert evaluator, were successful in anticipating several instances of actual mode confusion. The researcher then recommended that the heuristic evaluation for mode confusion mitigation be used as a companion procedure together with a usability study to provide for greater coverage of mode confusion scenarios.

The next and final chapter presents a discussion of these results, a summation of the material presented in §1 through §6, discusses the contributions of our research, and identifies potential areas for future work.

7

Future work and contributions

Although the results from the heuristic evaluation and usability study were encouraging and in-line with the predictions of the researcher, we believe that we have only highlighted a portion of the heuristics' abilities. As these heuristics can be used in formative and summative evaluations – whether early or late in the development cycle – the combination of directed use cases and heuristics have the potential to significantly reduce the effort and associated costs necessary to identify situations of mode confusion in future digital camera designs.

7.1 Discussion and future work

As a starting point for future work, the researcher would recommend the removal of several factors, introduced as a result of our limited evaluative resources, that may have curbed the effectiveness of the heuristics.

Firstly, the heuristic evaluation was performed by an individual whose exposure to the sample digital cameras was only by proxy; the evaluator was only permitted access to the manufacturer's instruction manuals and not the cameras themselves. The reason for this severe restriction was to simulate the circumstances a system engineer would face if working on a design for which a prototype had not yet been developed. As these heuristics are intended for consideration as early as possible in the design phase, it was important that the evaluator be given no advantage over a system engineer during the heuristic evaluation. In

many ways, the evaluator was at a greater disadvantage because they did not have the benefit of being intimately familiar with the intricacies of each camera design as the evaluator was not privy to the technical considerations and trade-offs made by the engineering team that a designer would have been, being an active participant in the system's development. Add to this the fact that the evaluator's level of expertise with each camera model was directly related to the quality of the supplied documentation – which was almost certainly abbreviated for the consumer – and it is clear that the ability for the evaluator to develop a fully realized understanding of the camera's design sensibilities, product goals, and technical operation was limited from the onset. The quality of this documentation is also paramount as any omissions or inaccuracies in the documentation would adversely distort the evaluator's understanding of the camera's functionality. If the evaluator was afforded the same insight and technical knowledge as a member of the development team for each of the sample cameras, it is surmised that a greater number of instances of mode confusion potential could have been catalogued.

Secondly, it must be appreciated that the results of the heuristic evaluation only represent the efforts of one individual. Nielsen believes that as few as five usability experts, using an introspective evaluation technique such as these heuristics, can uncover as much as 75% of a system's usability problems [9]. Just as the mental models of a system are very individualistic, the factors leading up to an instance of mode confusion likewise varies from user-to-user. If more than one evaluator were recruited to employ these heuristics, there would be a greater chance that personal variation could lead to a greater coverage of the potential mode confusion issues.

Thirdly, with respect to the composition of the user population, we acknowledge that there are several limitations regarding the diversity of the participants. Specifically, all but one of the subjects were male, almost all subjects were within the age bracket of twenty to thirty years of age, and all users were Computer Science students. While the limited sample size may not be

large enough to be considered statistically significant and include enough variation in composition to be considered broad, the fact that these well-educated users, with an established familiarity with systems requiring a high degree of technical ability, encountered usability issues underscores the fact that significant usability issues may exist in contemporary digital camera designs.

Finally, the three sample cameras that were reviewed are off-the-shelf products, representing designs that have had the benefit of a complete development lifecycle. As such, we can assume that each design has already undergone a battery of testing and refinement cycles by each respective engineering team before being released to manufacturing and distributed in retail channels. If the heuristics had been applied to “green” designs that had not yet seen refinement, it is conjectured that a significantly greater number of potential design flaws could have been uncovered using these heuristics.

As one of the target groups for these heuristics are designers of digital cameras with little or no expertise in human-computer interaction, an ideal scenario for future study would be to deploy the heuristics in the field as part of a digital camera design process and to conduct two usability studies: the first in which the participants are comprised of the designers themselves, to gain a better measure of the effort necessary to apply these heuristics; and a second with participants drawn from the target market of the camera to determine if any additional instances of mode confusion, not previously identified during the design process, can be found. Any additional instances of mode confusion found could then be used to revise the set of heuristics, further improving their ability to identify scenarios with mode confusion potential.

It has also been suggested that the heuristics could be modified into a *quasi* checklist, for use by consumers who are in the process of purchasing a digital camera, to determine

whether a particular model of digital camera has mode confusion potential. Issues identified by the consumer during *ad-hoc* usability testing could then be gauged to determine the severity of such issues and to decide whether such levels of confusion potential are acceptable. A change to the style of presentation of the heuristics and a refocus from designer to consumer would be likely modifications before deployment of the heuristics as a pre-purchase evaluation tool.

7.2 Research contributions

We believe that our research has made the following contributions to the field of human-computer interaction and more specifically, to the study of mode confusion identification and mitigation:

- A set of heuristics for identifying and mitigating mode confusion, written as a deliverable document. These heuristics, when used in conjunction with use cases in a formative evaluation, have the ability to predict and mitigate mode confusion potential in digital cameras.
- The collation of a generalized set of use cases for digital cameras that can be consulted by designers – during the conception phase of the product development cycle – to help establish a basic set of features, referenced during the instruction manual writing process to identify essential items for documentation, and incorporated into the testing process of future digital camera usability studies.
- The realization that most instances of mode confusion, at least within the sample set of digital cameras, occur in the more esoteric portions of digital camera designs, providing a possible focus for future usability studies.

- The groundwork for a modified version of these heuristics, suitable for use by consumers, to determine whether a particular model of digital camera has mode confusion potential before purchase.

7.3 Epilogue

Most of the usability issues found by our heuristic evaluation and usability study in our sample set of digital cameras occurred in the less commonly accessed portions of the user interface, suggesting that these areas may not have seen as much exposure during usability testing as their first-tier counterparts. The fact that instances of mode confusion remain in shipping products lends credence to the belief that many systems are released to manufacture before they have been thoroughly tested for usability issues, possibly due to market pressures and economic demands.

Perhaps it is this final matter of economics that underscores the need for a set of heuristics capable of mitigating mode confusion early in the design process. As the researcher's own experience has suggested that system engineers are largely unfamiliar with human-computer interaction principles and of usability testing techniques in general, much effort is focused on building the functionality of a system, leading to what Sarter and Woods has labelled "automation-centric computing" [44]. Usability issues are often contemplated in an ad-hoc fashion and too late in the development cycle to be properly addressed. Combine this with a common belief amongst some designers that usability problems can be resolved through an increased verbosity in documentation and it is clear that a solution – of which these heuristics only represent a basis – is needed to dismiss this pervasive and systemic inattention towards human factors.

References

- [1] A. Blackler, V. Popovic, and D. Mahar. Intuitive use of products. In *Proceedings of Design Research Society International Conference 2002*, London, pages 120–134, September 2002.
- [2] M. Brown and N. G. Leveson. Modelling controller tasks for safety analysis. In *Workshop on Human Error and System Development 1998*, Seattle, Washington, April 1998. URL: <http://sunnyday.mit.edu/papers/hessd98.pdf>
- [3] J. C. Campos and M. D. Harrison. From interactors to SMV: A case study in the automated analysis of interactive systems. Technical report YCS-99-317, Department of Computer Science, University of York, 1999.
- [4] Canon Corporation. *PowerShot A40 Digital Camera User Guide*. Tokyo, 2002.
- [5] Canon Corporation. *PowerShot S30 Digital Camera User Guide*. Tokyo, 2002.
- [6] J. Crow, D. Javaux, and J. Rushby. Models and mechanized methods that integrate human factors into automation design. In *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics (HCI-AERO 2000)*, Toulouse, France, pages 163–168, September 2000.
- [7] A. Degani. *Modeling Human-Machine Systems: On Modes, Errors, and Patterns of Interaction*. Ph. D. thesis. Georgia Institute of Technology, 1996.
- [8] A. Degani, C. M. Mitchell, and A. R. Chappell. Use of the operator function model to represent mode transition. In *Proceedings of the 8th International Symposium on Aviation Psychology*, Columbus, Ohio, pages 210–215, April 1995.

- [9] A. J. Dix, J. E. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*, second edition. Prentice Hall, 1998.
- [10] Eastman Kodak Company. *Kodak DC265 Zoom Digital Camera User's Guide*. Rochester, New York, 1999.
- [11] C. M. Harrison. A user-centred approach to digital camera design: Maximizing usability and facilitating acceptance of technology using psychological and ergonomic principles. In *Proceedings of the International Ergonomics Association XV Triennial Congress*, Seoul, August 2003. URL: http://www.hitlabnz.org/fileman_store/2003-IEA-UserCentredCameraDesign.pdf
- [12] R. Hourizi and P. Johnson. Unmasking mode errors: a new application of task knowledge principles to the knowledge gaps in cockpit design. In M. Hirose, editor, *Proceedings of the 8th IFIP Conference on Human Computer Interaction (Interact '01)*, Tokyo, July 2001. URL: <http://www.cs.bath.ac.uk/~hci/papers/Hourizi-Johnson-Interact-2001.pdf>
- [13] D. Javaux. Explaining Sarter & Woods' classical results: The cognitive complexity of pilot-autopilot interaction on the Boeing 737-EFIS. In *Proceedings of the 2nd IFIP Working Conference on Human Error, Safety, and System Development (HESSD 1998)*, Seattle, Washington, April 1998. URL: www.dcs.gla.ac.uk/~johnson/papers/seattle_hessd/javaux-p.pdf
- [14] B. E. John and D. E. Kieras. The GOMS family of user interface analysis techniques: comparison and contrast. In *ACM Transactions on Computer-Human Interaction*, volume 3, number 4, pages 320–351. ACM Press, 1996.
- [15] P. N. Johnson-Laird. Mental models and deduction. In *Trends in Cognitive Sciences*, volume 5, number 10, pages 434–442. Elsevier, 2001.
- [16] T. Jokela. When good things happen to bad products: Where are the benefits of usability in the consumer appliance market? In S. Dray and D. Siegel, editors, *Interactions*, volume 11, number 6, pages 28–35. Lawrence Erlbaum, 2004.
- [17] S. R. Kang. User experience: Beyond usability. In *Bulletin of the 5th Asian Design Conference / International Symposium on Design Science*, Seoul, pages 121–125, October 2003.

- [18] P. Ketola and M. Røykkee. The three facets of usability in mobile handsets. In *CHI 2001, Workshop, Mobile Communications: Understanding Users, Adoption & Design*, Seattle, Washington, April, 2001. URL: http://www.cs.colorado.edu/~palen/chi_workshop/papers/ketola.pdf
- [19] J. A. Landay and T. R. Kaufmann. User interface issues in mobile computing. In *Proceedings of the Fourth Workshop on Workstation Operating Systems*, Napa, California, pages 40–47, October 1993.
- [20] A. Lankenau. Avoiding mode confusion in service robots: The Bremen autonomous wheelchair as an example. In *Proceedings of the 7th International Conference On Rehabilitation Robotics*, Evry, France, April 2001. URL: <http://www.informatik.uni-bremen.de/kogrob/papers/icorr2001.pdf>
- [21] N. G. Leveson, R. J. Hansman, M-A. D. Storey. Approaches to human-centered software development. 2002.
- [22] N.G. Leveson and E. Palmer. Designing automation to reduce operator errors. In *Proceedings of Systems, Man, and Cybernetics Conference*, Orlando, Florida, October 1997. URL: <http://sunnyday.mit.edu/papers/smc.pdf>
- [23] N.G. Leveson, L. D. Pinnel, S. D. Sandys, S. Koga, and J D. Reese. Analyzing software specifications for mode confusion potential. In *Workshop on Human Error and System Development 1997*, Glasgow, March 1997. URL: <http://sunnyday.mit.edu/papers/glasgow.pdf>
- [24] C. Lewis, P. Polson, C. Wharton, and J. Rieman. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proceedings of Conference on Human Factors in Computing Systems (CHI '90)*, Seattle, Washington, pages 235–242, April 1990.
- [25] G. Lüttgen and V. Carreño. Analyzing mode confusion via model checking. Technical report 99-18, Institute for Computer Applications in Science and Engineering, NASA, May 1999.
- [26] R. Molich and J. Nielsen. Improving a human-computer dialogue. In E. H. Sibley, editor, *Communications of the ACM: Computing Practices*, volume 33, number 3. ACM Press, 1990.

- [27] N. A. Neogi. *Hazard Elimination Using Backwards Reachability Techniques In Discrete and Hybrid Models*. Ph. D. thesis. Massachusetts Institute of Technology, 2002.
- [28] J. Nielsen. Reliability of severity estimates for usability problems found by heuristic evaluation. In *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems*, Monterey, California, pages 129–130, May 1992.
- [29] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of Conference on Human Factors in Computing Systems (CHI '90)*, Seattle, Washington, pages 249–256, April 1990.
- [30] D. A. Norman. *The Design of Everyday Things*. Doubleday, 1990.
- [31] D. A. Norman. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer is So Complex, and Information Appliances are the Solution*. MIT Press, 1999.
- [32] W. A. Olson. Identifying and mitigating the risks of cockpit automation. Air Command Staff College, United States Air Force, 2001. URL: http://aupress.au.af.mil/Wright_Flyers/Text/wf14.pdf
- [33] W. A. Olson and N. B. Sarter. Management-by-consent in human-machine systems: When and why it breaks down. In N.J. Cooke editor, *Human Factors: The Journal of the Human Factors and Ergonomics Society*, volume 43, number 2, pages 255–266. Human Factors and Ergonomics Society, 2001.
- [34] E. Palmer. “Oops, it didn’t arm” – a case study of two automation surprises. Technical report, Ames Research Center, NASA, 1996.
- [35] S. Pocock, M. Harrison, P. Wright, and P. Johnson. THEA: A technique for human error assessment early in design. In M. Hirose, editor, *Proceedings of the 8th IFIP Conference on Human Computer Interaction (Interact '01)*, Tokyo, pages 247–254, July 2001.
- [36] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey. *Human-Computer Interaction*. Addison-Wesley, 1994.

- [37] J. Rasmussen. Designing to support adaptation. In *Proceedings of the Joint 14th Triennial Congress of the International Ergonomics Association and the 44th Annual Meeting of the Human Factors and Ergonomics Society (IEA2000/HFES2000)*, Santa Monica, California, July 2000.
- [38] M. Rodriguez, M. Zimmerman, M. Katahira, M. Villepin, B. Ingram, and N. G. Leveson. Identifying mode confusion potential in software design. In *Proceedings of the 19th Digital Avionics and Systems Conference (DASC 2000)*, Philadelphia, Pennsylvania, October 2000.
- [39] J. Rushby, J. Crow, and E. Palmer. An automated method to detect potential mode confusions. In *Proceedings of the 18th AIAA/IEEE Digital Avionics Systems Conference (DSAC 1999)*, St. Louis, Missouri, October 1999.
- [40] N. B. Sarter and B. Schroeder. Supporting decision-making and action selection under time pressure and uncertainty: The case of in-flight icing. In N.J. Cooke editor, *Human Factors: The Journal of the Human Factors and Ergonomics Society*, volume 43, number 4, pages 573–583. Human Factors and Ergonomics Society, 2001.
- [41] N. B. Sarter and D. D. Woods. Pilot interaction with cockpit automation. I. Operational experiences with the flight management system. In *International Journal of Aviation Psychology*, volume 2, number 4, pages 303–321. Lawrence Erlbaum, 1992.
- [42] N. B. Sarter and D. D. Woods. Pilot interaction with cockpit automation. II. An experimental study of pilot's mental model and awareness of the flight management system (FMS). In *International Journal of Aviation Psychology*, volume 4, number 1, pages 1–28. Lawrence Erlbaum, 1994.
- [43] N. B. Sarter and D. D. Woods. Strong, silent, and 'out-of-the-loop'. CSEL report 95-TR-01, Ohio State University, 1995.
- [44] L. Sherry, P. Polson, and M. Feary. Designing user-interfaces for the cockpit: Five common design errors and how to avoid them. In *Proceedings of the 2002 SAE World Aviation Congress*, Phoenix, Arizona, November 2001.
- [45] D. D. Woods and E. S. Patterson. How unexpected events produce an escalation of cognitive and coordinative demands. In P.A. Hancock and P.A. Desmond, editors, *Stress, Workload and Fatigue*, pages 290–304. Lawrence Erlbaum, 2001.

-
- [46] D. D. Woods, E. S. Patterson, E. M. Roth, and K. Christoffersen. Can we ever escape from data overload? In *Cognition, Technology & Work*, volume 4, number 1, pages 22–36. Springer, 2002.
- [47] G. van der Veer and H. van Vliet. The human-computer interface is the system; a plea for a poor man’s HCI component in software engineer curricula. In *Proceedings of the 14th Conference on Software Engineering Education and Training*, Charlotte, North Carolina, pages 276–288, February 2001.
- [48] University of Victoria, Human Research Ethics Board. Mission statement. URL: <http://www.research.uvic.ca/ethics/ethicsmain.htm>



User study: certificate of approval



University
of Victoria

University of Victoria - Human Research Ethics Committee

Certificate of Approval

<u>Principal Investigator</u> Victor Chong Graduate Student <u>Co-Investigator(s)</u> :	<u>Department/School</u> COSI	<u>Supervisor</u> Dr. Margaret-Anne Storey	
Title: Observation of Users Interacting with Digital Cameras to Identify Scenarios with Mode Confusion Potential			
<u>Project No.</u> 388-03	<u>Approval Date</u> 07-Oct-03	<u>Start Date</u> 15-May-03	<u>End Date</u> 06-Oct-04

Certification

This is to certify that the University of Victoria Ethics Review Committee on Research and other Activities Involving Human Subjects has examined the research proposal and concludes that, in all respects, the proposed research meets appropriate standards of ethics as outlined by the University of Victoria Research Regulations Involving Human Subjects.

J. Howard Brunt
Associate Vice-President, Research

This Certificate of Approval is valid for the above term provided there is no change in the procedures. Extensions/minor amendments may be granted upon receipt of "Request for Continuing Review or Amendment of an Approved Project" form.

Office of Vice-President, Research - UVic
Room 424 BEC, P.O. Box 1722
Victoria, BC V8W 2Y2

Tel: (250) 472-4362
Fax: (250) 721-8960
E-mail: ovpres@uvic.ca

388-03 Chong, Victor



Usability study: questionnaire

Questionnaire

Observation of Users Interacting with Digital Cameras to Identify Scenarios with Mode Confusion Potential

You have been invited to participate in a study entitled “Observation of Users Interacting Digital Cameras to Identify Scenarios with Mode Confusion Potential” that is being conducted by **Victor Chong**. Victor Chong is a graduate student in the Department of Computer Science at the University of Victoria and you may contact him if you have questions at **(250) 472-4239** or **vchong@cs.uvic.ca**.

A framework has been developed to identify scenarios where a potential for mode confusion exists in the user interfaces of digital cameras. A set of digital cameras have been evaluated using this framework and tasks with mode confusion potential have been identified. The purpose of this study is to observe users interacting with these digital cameras to identify real-world instances of mode confusion and, by comparing these with the set of tasks identified by the framework, evaluate the framework’s ability to anticipate situations of mode confusion.

This questionnaire is to be completed following the experiment as a means of recording your experience the digital cameras used over the course of this study. You are not obligated to answer any of the questions and may withdraw from the questionnaire at any time. If you do withdraw from the questionnaire, your data will be retained and used in the study if you agree, otherwise you may choose to exclude some or all of your contributions to the questionnaire.

This questionnaire consists of **6 questions**, arranged in **3 sections**. Space is provided at the end of the questionnaire for you to record any additional comments that you feel would be of benefit to the investigator. The estimated time to complete this questionnaire is **15 minutes**. If you have any questions regarding the questionnaire, please feel free to ask the investigator.

1.0 Background

1.1 How many years of experience do you have with digital cameras?

- | | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| < 1 year | 1 to 2 years | 3 to 4 years | 4 to 5 years | > 5 years |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

1.2 How would you rate your familiarity with digital cameras?

- | | | | | |
|---------------|----------|---------------------|----------|---------------|
| novice | | intermediate | | expert |
| 1 | 2 | 3 | 4 | 5 |

2.0 Your Experience with the Digital Cameras

2.1 While performing the requested tasks, I encountered difficulties with the Canon PowerShot A40.

- | | | |
|--------------------------|--------------------------|--------------------------|
| yes | no | uncertain |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

If you answered "Yes", please state the nature of your difficulty, and any resolution, below:

2.2 While performing the requested tasks, I encountered difficulties with the Canon PowerShot S30.

- | | | |
|--------------------------|--------------------------|--------------------------|
| Yes | No | uncertain |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

If you answered "Yes", please state the nature of your difficulty, and any resolution, below:

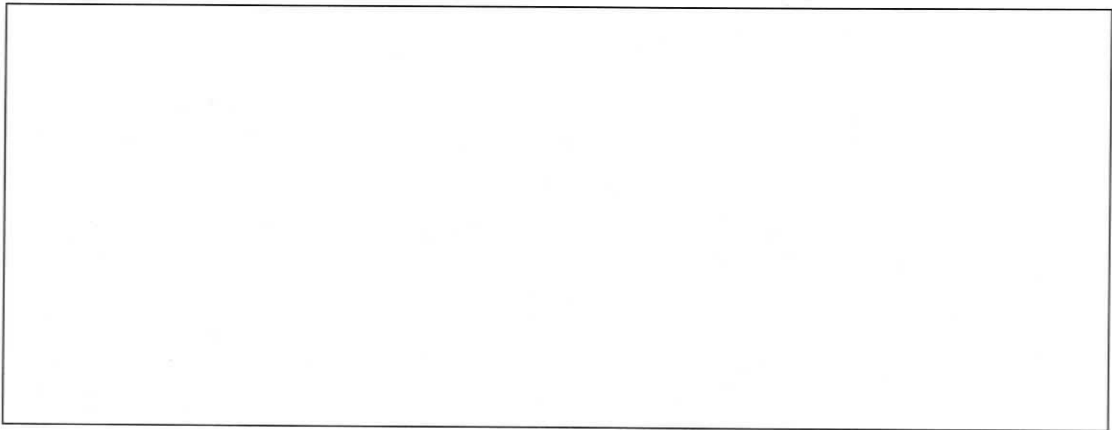
2.3 While performing the requested tasks, I encountered difficulties with the Olympus C-3040 Zoom.

- | | | |
|--------------------------|--------------------------|--------------------------|
| yes | no | uncertain |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

If you answered "Yes", please state the nature of your difficulty, and any resolution, below:

3.0 Your Model Mode of the Canon PowerShot S30's Operating Modes

In the space below, please draw a diagram showing the relationship between the sliding playback button, the light immediately to the left of the button, and the sliding lens cover. Feel free to play with the S30 to help you with this task.



Additional comments

Thank-you for completing the questionnaire. Please use the space below to record any additional comments or observations that you feel would be of benefit to the investigator.



User study: Canon PowerShot S30 sliding lens cover

Perhaps the most interesting aspect of the post-session questionnaire was the variety of answers provided for the third section, “Your Mental Model of the Canon PowerShot S30’s Operating Modes.” In this part of the questionnaire, the subjects were asked to draw a state transition diagram to show the relationship between two actuators, one indicator, and the operating modes on the Canon PowerShot S30.

As mentioned in the previous section, we were particularly interested to see if there existed a disparity between the user’s mental model of the system and the actual system model as designed by the engineers [30]. As the state transition diagram is a commonly taught visualization technique in the computing sciences, state transition diagrams were a convenient method for users to record the perceived interactions between the two controls, indicator, and operating modes of the Canon PowerShot S30.

C.1 System model as designed by the manufacturer

The following is an excerpt from the operational guide shipped with the Canon PowerShot S30 camera. This diagram accurately expresses the relationship between the lens cover position and operating modes.

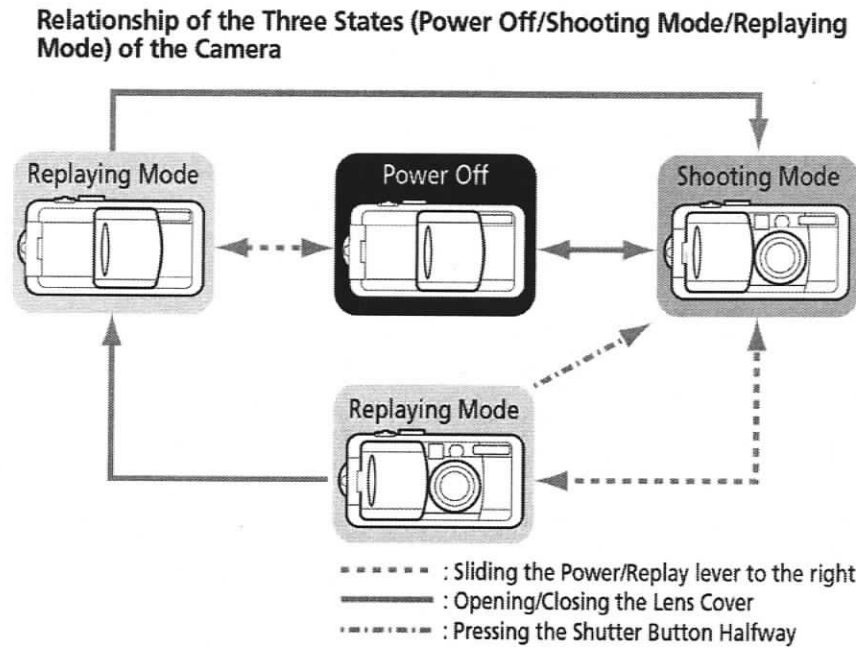


Figure C.1: Relationship of the Canon PowerShot S30's lens cover position and operating mode as provided by the camera manufacturer.

C.2 Mental models as developed by the users

The following are reproductions of the concept diagrams drawn by the users during the observation session. Users were asked to graphically document the relationship between the sliding lens cover position and the operating modes of the Canon PowerShot S30 digital camera.

Participant A

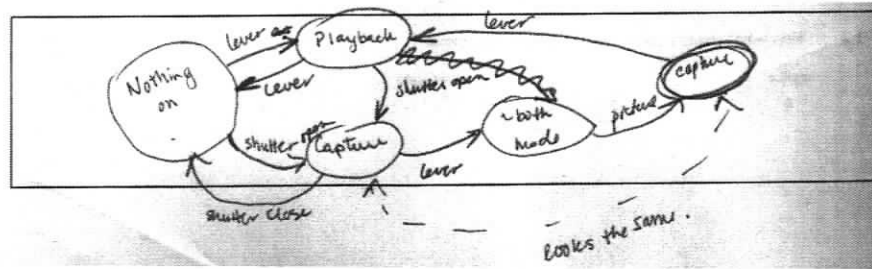


Figure C.2a: Relationship of lens cover position and operating mode as understood by Participant A. Of particular interest is this participant's belief that there exists two distinct capture modes which look similar (dashed line with "looks the same" comment), but behave differently.

Participant B

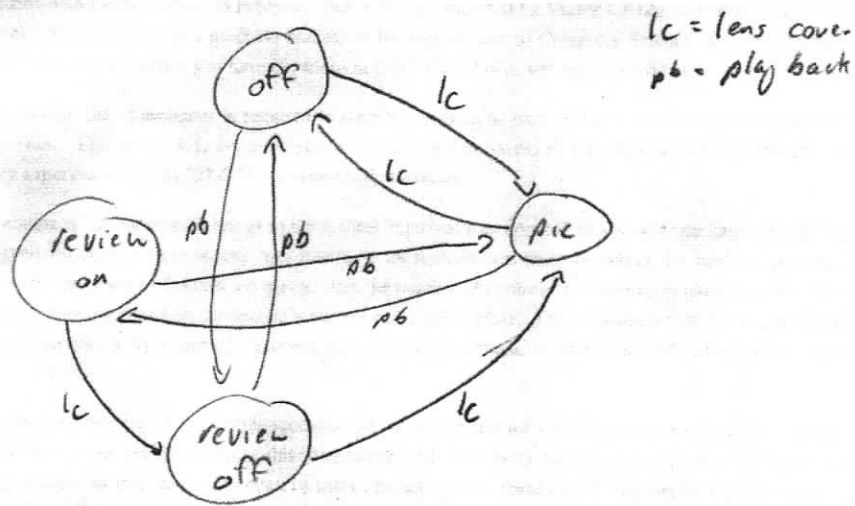


Figure C.2b: Relationship of lens cover position and operating mode as understood by Participant B. This mental model is mostly accurate except that it not express the mode of the S30 wherein the camera is in capture mode and the LCD display is being used as a live preview.

Participant C

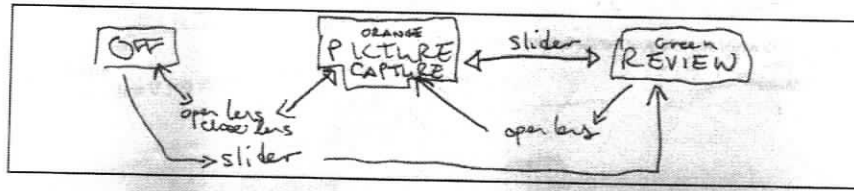


Figure C.2c: Relationship of lens cover position and operating mode as understood by Participant C. This diagram is mostly complete except for a missing mode transition that occurs when the lens cover is closed from the [lens cover open, review images] condition.

Participant D

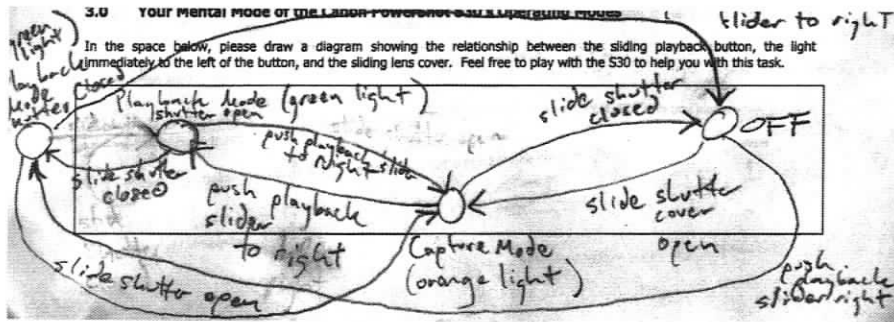


Figure C.2d: Relationship of lens cover position and operating mode as understood by Participant D.

Participant E

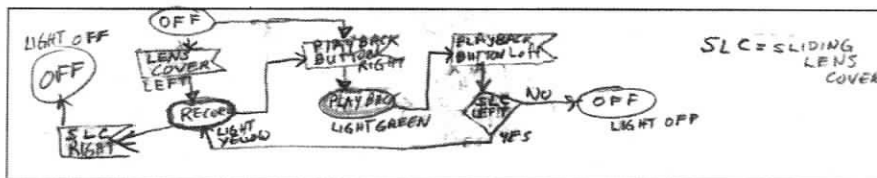


Figure C.2e: Relationship of lens cover position and operating mode as understood by Participant E. Like Participant C, this mental model is missing the transition that occurs when the camera's sliding lens cover moved from the open to closed position while the camera is in image review mode.

Participant F

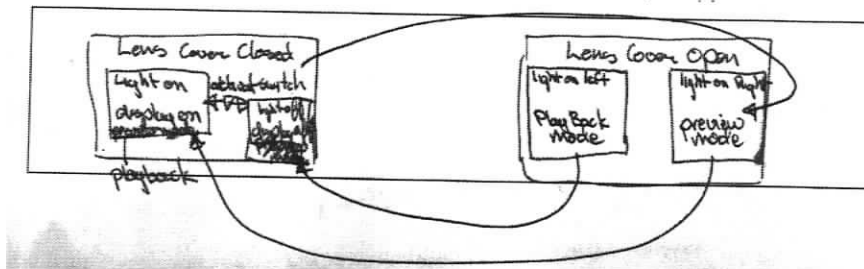


Figure C.2f: Relationship of lens cover position and operating mode as understood by Participant F. This mental diagram does not make any mention of the playback slider switch but does refer to a “light on left” and “light on right” which, unfortunately, does not seem to correlate to any feature on the Canon PowerShot S30.

Participant G

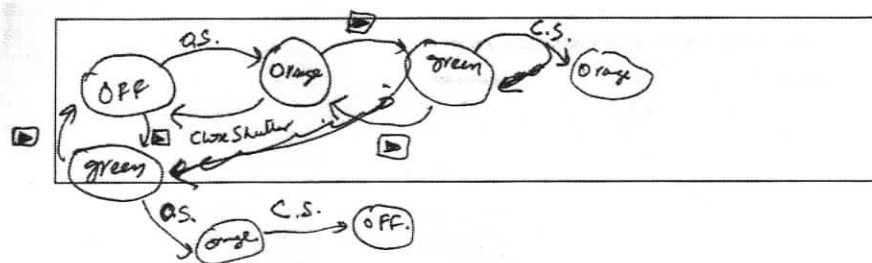


Figure C.2g: Relationship of lens cover position and operating mode as understood by Participant G. Although there is a slight mistake in the drawing of the diagram (the right-most “orange” mode is in fact the same as the left-most “orange” mode), this state transition diagram is correct.

C.3 Discussion

Of the mental models provided by the users, Participant G’s state transition diagram is the closest match to the actual system as designed by the manufacturer. If an allowance is made to correct this minor transcription error, only one participant was able to accurately describe the behaviour of the Canon PowerShot S30.

This finding is particularly interesting considering that every advantage has been given to this set of users: the study participants are graduate students in the field of Computer Science who are accustomed to recognizing the logical behaviour of systems and are trained in the use of state transition diagrams to accurately document mode transitions. If users of such a calibre are unable to truthfully reproduce the behaviour of the PowerShot S30, it may prove even more difficult for a typical user without such sensibilities to properly understand the complex interaction between the sliding lens cover, the playback switch, and the bi-colour indicator light.